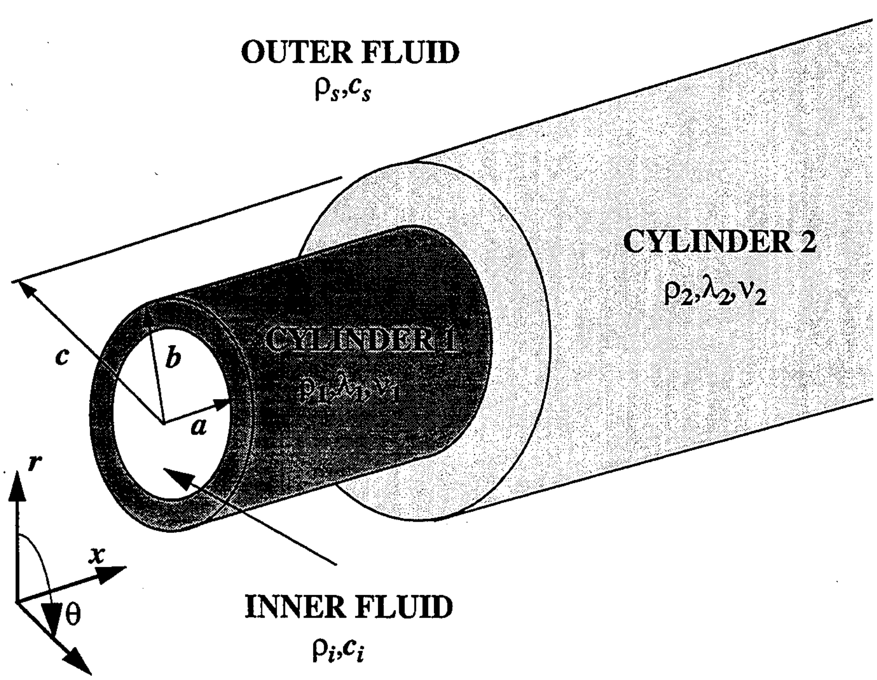
NUWC-NPT Technical Document 11,065
29 November 1995

FORTRAN Algorithms for the Three-Dimensional Solution of Two-Layer Solid and Hollow Cylinder Dynamic Elasticity Problems With and Without Fluids

Mark S. Peloquin
Submarine Sonar Department



Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

The research presented in this document was sponsored by the Ocean, Atmosphere, and Space S&T Department, Sensing and Systems Division, of the Office of Naval Research (ONR), Program Element 0602314N. The ONR Program Manager for Sensors, Sources, and Arrays is K. G. Dial (ONR 321SS).

The author wishes to thank K. A. Holt for her help with the editing of the manuscript.

Reviewed and Approved: 29 November 1995

R. J. Martin
Acting Head, Submarine Sonar Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	29 November 1995	Final

4. TITLE AND SUBTITLE

FORTRAN Algorithms for the Three-Dimensional Solution of Two-Layer Solid and Hollow Cylinder Dynamic Elasticity Problems With and Without Fluids

5. FUNDING NUMBERS

6. AUTHOR(S)

Mark S. Peloquin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center
Detachment New London
New London, Connecticut 06320

8. PERFORMING ORGANIZATION REPORT NUMBER

TD 11,065

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 No. Quincy Street
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This document contains FORTRAN algorithms for the numerical evaluation of exact closed form solutions to four fluid/structure interaction mathematical problems that are modeled as cylinders and fluids. The cylinders (infinitely long in the longitudinal *x*-direction) are modeled as three-dimensional linear viscoelastic solids, and the fluids are treated as inviscid. The outer fluid extends to infinity in the radial *r*-direction; the inner fluid (contained within the hollow cylinder) is bounded by the inner cylinder at radius $r = a$. The cylinders are subject to forced harmonic vibration at the solid/fluid interface. The response to both axisymmetric and nonaxisymmetric excitations is calculated, with nonaxisymmetric excitation described in terms of circumferential order number n.

The first model is comprised of a single hollow cylinder, filled with an inner fluid and immersed in an outer fluid. The second model consists of a single solid cylinder immersed in an outer fluid. The third model, an extension of the second one, adds a second cylinder in contact with the first cylinder. The fourth model consists of the two-layered cylinder with outer fluid (model three); however, the inner cylinder is now hollow and contains an inner fluid. Structural damping requires that the material properties be represented as complex numbers; therefore, series solutions to the Bessel functions are required. Native algorithms are provided for the Bessel functions, with certain parameter restrictions, and a subroutine is given to allow use of the external FORTRAN Bessel function libraries provided by MathWorks, Inc., without parameter restriction.

14. SUBJECT TERMS

Closed Form Solution, Cylinder, Dynamic Elasticity, Excitation, Fluid/Structure Interaction, Forced Harmonic Vibration, FORTRAN Algorithms, Three-Dimensional Solution

15. NUMBER OF PAGES

260

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

LIST OF ILLUSTRATIONS

LIST OF TABLES

FORTRAN ALGORITHMS FOR THE THREE-DIMENSIONAL SOLUTION OF TWO-LAYER SOLID AND HOLLOW CYLINDER DYNAMIC ELASTICITY PROBLEMS WITH AND WITHOUT FLUIDS

INTRODUCTION

Closed form solutions for wave propagation in a two-layered cylinder with outer fluid loading were analyzed in NUWC-NPT Technical Report 11,043.[1] The solution for the subproblem consisting of a single solid cylinder in contact with an outer fluid was also presented in this reference.

Additionally, a closed form solution for wave propagation in a two-layered cylinder with inner and outer fluid loading was derived in NUWC-NPT Technical Report 11,067.[2] The subproblem of a single solid cylinder in contact with inner and outer fluids was also addressed.

In both of these previous derivations, the cylinders were infinite in the longitudinal direction. Damping was modeled using a complex modulus of elasticity, which required a series solution for the requisite Bessel functions. Three excitation states were considered: the normal pressure P_o, the longitudinal shear stress P_x, and the circumferential shear stress P_θ. The response was derived for nonaxisymmetric excitation in terms of circumferential order number n. The axisymmetric response was obtained by the degenerate case of $n = 0$.

The closed form solutions described above have been programmed in FORTRAN to obtain numerical results. This document details the FORTRAN algorithms developed to describe the cases of wave propagation presented in references 1 and 2. Variable lists, subroutine

---

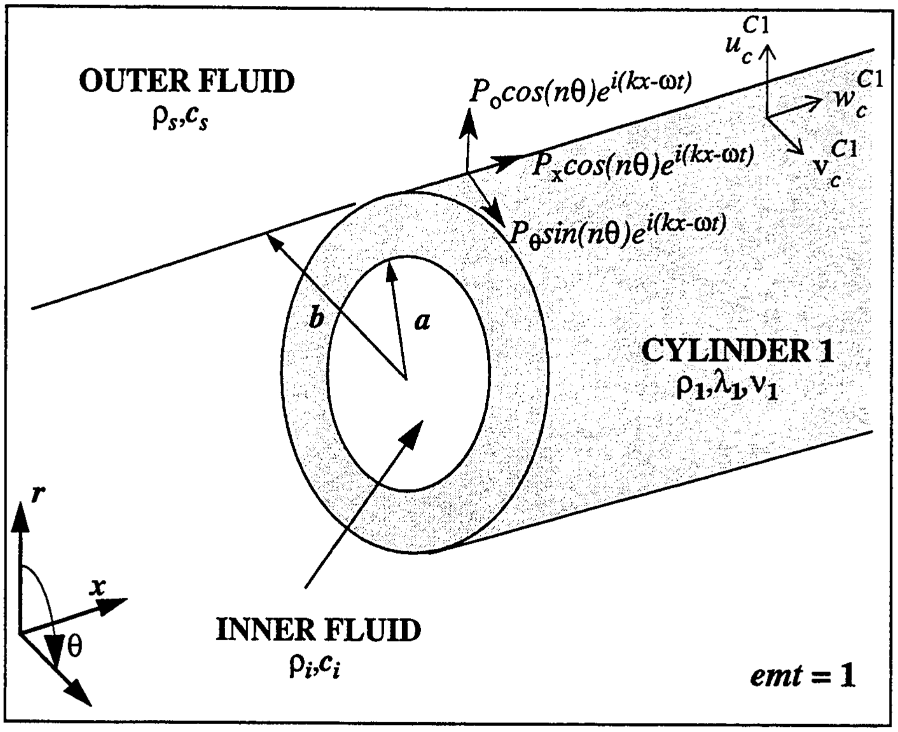
1. M. S. Peloquin, "A Three-Dimensional Elasticity Solution for Wave Propagation in a Two-Layered Infinite Viscoelastic Solid Cylinder With Outer Fluid Loading," NUWC-NPT Technical Report 11,043, Naval Undersea Warfare Center Detachment, New London, CT, 31 August 1995.
2. M. S. Peloquin, "A Closed Form Dynamic Elasticity Solution to the Fluid/Structure Interaction Problem of a Two-Layered Infinite Viscoelastic Cylinder With Inner and Outer Fluid Loading Subject to Forced Harmonic Excitation," NUWC-NPT Technical Report 11,067, Naval Undersea Warfare Center Detachment, New London, CT, 30 December 1995.

descriptions, flow charts, and a cross reference of the FORTRAN program are presented.

The input variables are contained in tables 1 through 6. These tables relate the FORTRAN variables to the corresponding symbols used in the references. Intermediate variables used in the FORTRAN algorithms and the references are listed in appendixes A and B. Table 7 lists the output variables necessary to specify the quantity desired for output and the excitation type. It also lists the particular model to be used in the calculation:

- inner fluid/cylinder 1/outer fluid
- solid cylinder/outer fluid
- solid cylinder/cylinder 2/outer fluid
- inner fluid/cylinder 1/cylinder 2/outer fluid.

These four model types are identified by variable *emt* and are depicted in figures 1 and 2. Table 8 describes the output variable.



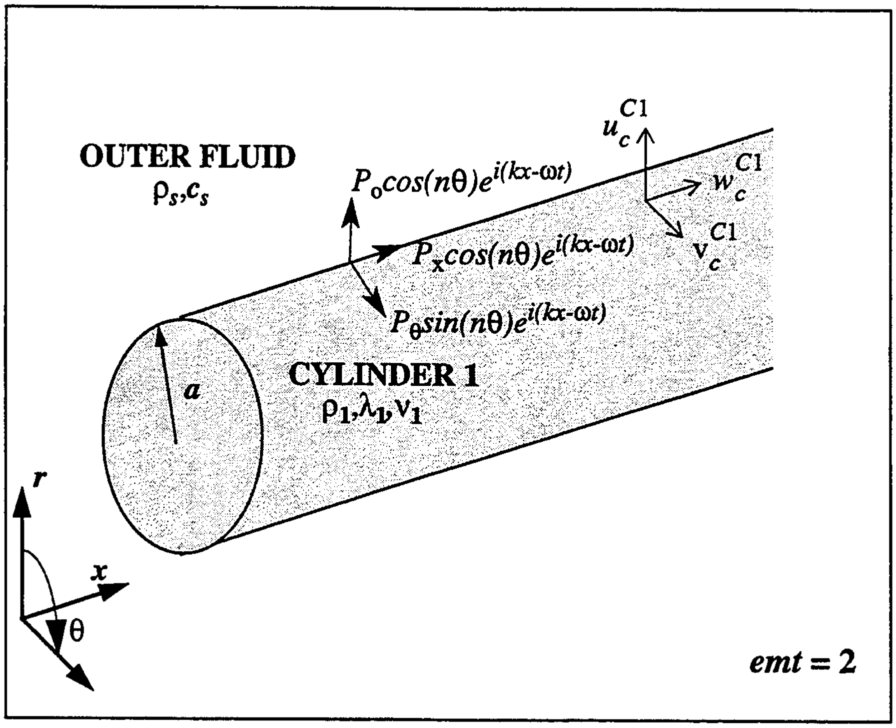
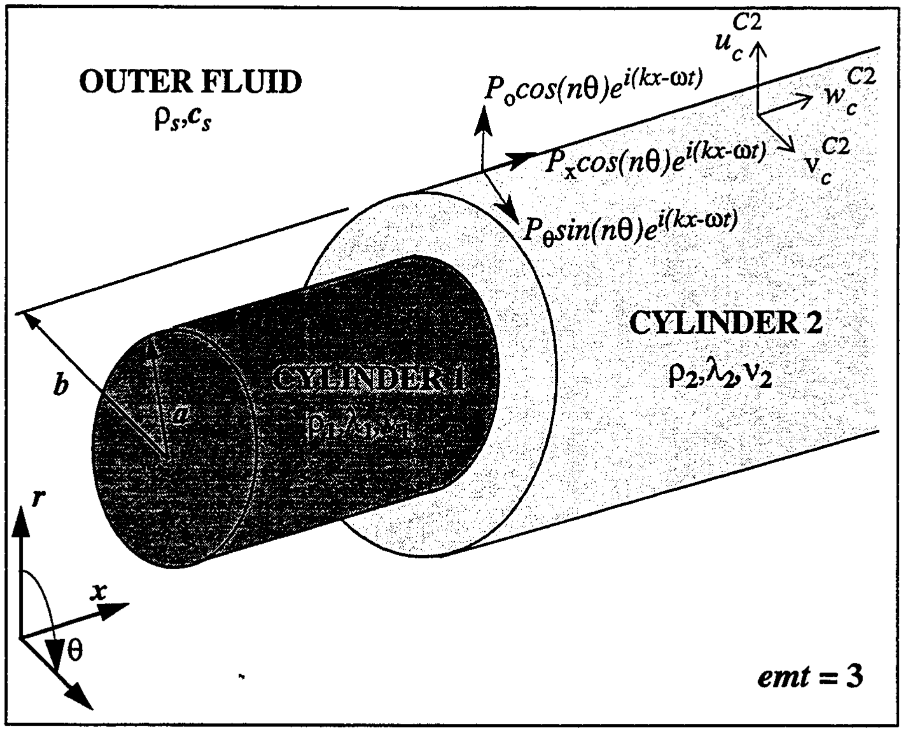


Figure 1. Model Types *emt* = 1 and *emt* = 2



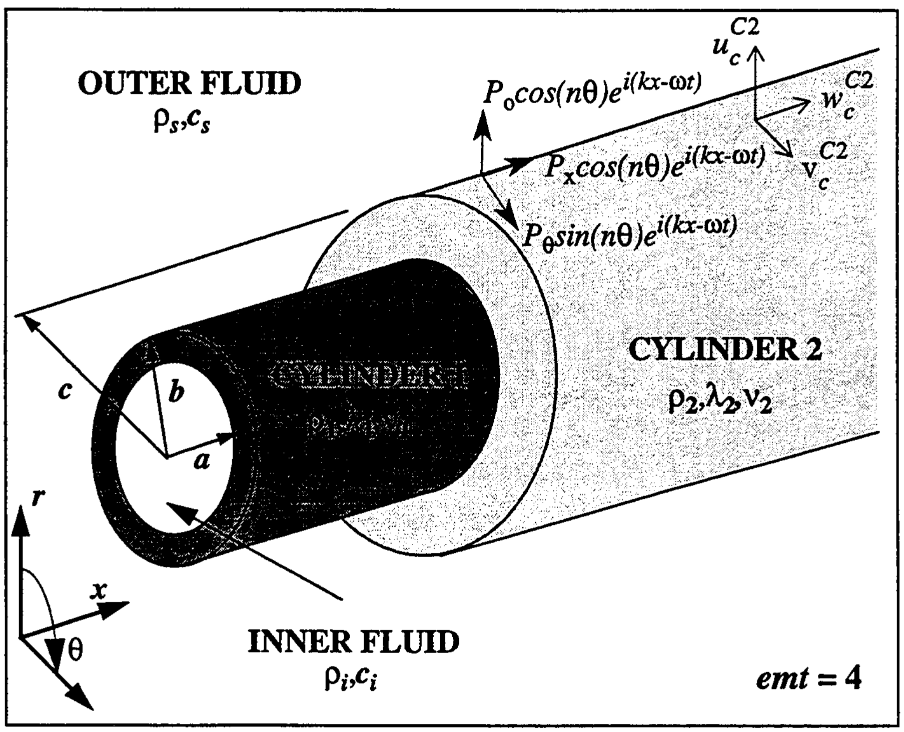


Figure 2. Model Types *emt* = 3 and *emt* = 4

INPUT/OUTPUT/VARIABLES

SOLID CYLINDER

Table 1. Inputs for the Solid Cylinder

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
E_rod	E_1	Young's modulus
zeta_rod	ζ_1	Structural loss factor
mu_rod	ν_1	Poisson ratio
r_rod	ρ_1	Density
ao_rod	a	Outer radius

CYLINDER 1

Table 2. Inputs for Cylinder 1

FORTRAN VARIABLE	TR 11,043 VARIABLE	TR11,067 VARIABLE	DESCRIPTION
E_1cyl	E_2	E_1	Young's modulus
zeta_1cyl	ζ_2	ζ_1	Structural loss factor
mu_1cyl	ν_2	ν_1	Poisson ratio
r_1cyl	ρ_2	ρ_1	Density
ao_1cyl	a	a	Inner radius
h_1cyl	No equivalent	No equivalent	Cylinder wall thickness

CYLINDER 2

Table 3. Inputs for Cylinder 2

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
E_2cyl	E_2	Young's modulus
zeta_2cyl	ζ_2	Structural loss factor
mu_2cyl	ν_2	Poisson ratio
r_2cyl	ρ_2	Density
ao_2cyl	b	Inner radius
h_2cyl	No equivalent	Cylinder wall thickness

OUTER FLUID

Table 4. Inputs for the Outer Fluid

FORTRAN VARIABLE	TR 11,043 TR11067 VARIABLE	DESCRIPTION
co	c_s	Speed of sound in the outer fluid
ro	ρ_s	Density of the outer fluid

INNER FLUID

Table 5. Inputs for the Inner Fluid

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
ci	c_i	Speed of sound in the inner fluid
ri	ρ_i	Density of the inner fluid

GENERAL VARIABLES

Table 6. General Variables

FORTRAN VARIABLE	TR 11,043 TR 11,067 VARIABLES	DESCRIPTION
k	k	Calculation wavenumber
Om	ω	Calculation frequency
r	r_1	Calculation radius
n	n	Circumferential order number

OUTPUT VARIABLES

Table 7. Output Setup Variables

FORTRAN VARIABLE	VALUE	DESCRIPTION
tft	0	RADIAL STRESS or FLUID PRESSURE
	1	LONGITUDINAL STRESS
	2	AXIAL DISPLACEMENT
	3	THETA DISPLACEMENT
	4	RADIAL DISPLACEMENT
	5	LONGITUDINAL DISPLACEMENT
	6	CIRCUMFERENTIAL STRAIN
	7	RADIAL STRAIN
	8	
	9	OPTICAL PHASE SENSITIVITY
	10	OPTICAL PHASE SENSITIVITY
exctype	0	LONGITUDINAL EXCITATION
	1	RADIAL EXCITATION
	2	CIRCUMFERENTIAL EXCITATION
emt	1	INNER FLUID/CYLINDER 1/OUTER FLUID
	2	SOLID CYLINDER/OUTER FLUID
	3	SOLID CYLINDER/CYLINDER 2/FLUID
	4	INNER FLUID/CYLINDER 1/CYLINDER 2/OUTER FLUID

Table 8. Output Variable

FORTRAN VARIABLE	VALUE	DESCRIPTION
g(jk)	Calculated	Contains the calculated quantity, specified by variables *tft* and *exctype*.

SUBPROGRAM DESCRIPTION

SUBPROGRAM mr2cf.f

This is a miniature main program that is used to evaluate the various wave propagation models described earlier. It is an interactive program in which the user enters the wavenumber and frequency at which the calculation will be performed. All the input and output variables are specified as well. Appendix C contains the makefile that was used to generate the executable.

The logical flow in the flow charts can be embedded in a routine that will generate an entire response surface over a range of wavenumbers and frequencies. This is the method used to generate the images in references 1 and 2.

SUBPROGRAM cbessl.f

The series solutions for the Bessel functions of complex argument used in reference 1 were generated with the functions contained in *cbessl.f* (*emt* = 2 and *emt* = 3). This group of functions is adequate for use when $n = 0$ or 1 for model *emt* = 1 or *emt* = 4. When n is greater than 1, the subroutine *cbessl.f_matlab* (appendix D) is used to call series solutions to the Bessel functions provided by MATLAB in its user libraries. The MATLAB user libraries are not provided in this document. Any Bessel function library could be used by making appropriate changes in the *cbessl.f_matlab* file and linking the Bessel function library of choice when compiling.

The following functions are contained in the FORTRAN subprogram *cbessl.f*:

Function gamma(n)

Gamma function of argument n.

Function fac(n)

Factorial of argument n.

Function psi(n)

Psi function of argument *n*.

Double Complex Function cbessj(n,a,r)

Bessel function of the first kind of integer order *n* and argument *ar*, where *a* is a complex number.

Double Complex Function cbessy(n,a,r)

Bessel function of the second kind of integer order *n* and argument *ar*, where *a* is a complex number.

Double Complex Function cbessk(n,a,r)

Modified Bessel function of integer order *n* and argument *ar*, where *a* is a complex number.

Double Complex Function d1cbessk(n,a,r)

First derivative with respect to *r* of the modified Bessel function *cbessk(n,a,r)*.

Double Complex Function d2cbessk(n,a,r)

Second derivative with respect to *r* of the modified Bessel function *cbessk(n,a,r)*.

Double Complex Function d1cbessj(n,a,r)

First derivative with respect to *r* of the Bessel function of the first kind *bessj(n,a,r)*.

Double Complex Function d2cbessj(n,a,r)

Second derivative with respect to *r* of the Bessel function of the first kind *bessj(n,a,r)*.

Double Complex Function d1cbessy(n,a,r)

First derivative with respect to *r* of the Bessel function of the second kind *bessy(n,a,r)*.

Double Complex Function d2cbessy(n,a,r)

Second derivative with respect to *r* of the Bessel function of the second kind *bessy(n,a,r)*.

Double Complex Function d1cbessi(n,a,r)

First derivative with respect to *r* of the modified Bessel function *cbessi(n,a,r)*.

Double Complex Function d2cbessi(n,a,r)

Second derivative with respect to *r* of the modified Bessel function *cbessi(n,a,r)*.

Double Complex Function cbessh1(n,a,r)

Hankel function of the first kind of integer order *n* and argument *ar*, where *a* is a complex number.

Double Complex Function cbessh2(n,a,r)

Hankel function of the second kind of integer order *n* and argument *ar*, where *a* is a complex number.

Double Complex Function d1cbessh1(n,a,r)

First derivative with respect to *r* of the Hankel function of the first kind *cbessh1(n,a,r)*.

Double Complex Function d1cbessh2(n,a,r)

First derivative with respect to *r* of the Hankel function of the second kind *cbessh2(n,a,r)*.

SUBPROGRAM rf.f

Subroutine ROD_POT(r,k,Om,n,cl_rod,ct_rod)

Displacement potentials for the solid cylinder are calculated. This subroutine is called once at $r = a$ in order to facilitate the solution of the undetermined coefficients. It is called a second time in cases *emt* = 2 and 3 when the calculation radius falls within the envelope of the solid cylinder. The following variables are calculated and returned by this subroutine in Common Block /ROD/:

```
C DEFINITION FOR COMMON BLOCK /ROD/
        complex*16 SP_rod,d1_SP_rod,d2_SP_rod
        complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
        complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
```

Subroutine SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)

Coefficients of the system matrix, *sm*, for the solid cylinder/outer fluid combination (*emt* = 2) are calculated. A fully populated *sm* matrix is returned.

Subroutine ABC_ROD_SOLVE(exctype,sm)

This subroutine solves for the undetermined coefficients of the solid cylinder/outer fluid combination (*emt* = 2) by using the method of determinants to invert the system matrix *sm*. This subroutine returns the following variables in Common Blocks /ROD/ and /OFLUID/:

```
C DEFINITION FOR COMMON BLOCK /ROD/
        complex*16 A1_rod,B1_rod,C1_rod

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        complex*16 M_OF
```

Subroutine ABC_ROD_INVERT(exctype,sm)

Undetermined coefficients are solved for through the inversion of matrix *sm* (*emt* = 2). The subroutine *MINV* is called from *ABC_ROD_INVERT* to perform the complex matrix inversion on matrix *sm*. This subroutine returns the following variables in Common Block /ROD/ and /OFLUID/:

```
C DEFINITION FOR COMMON BLOCK /ROD/
        complex*16 A1_rod,B1_rod,C1_rod

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        complex*16 M_OF
```

Subroutine OUTPUT(tft,n,k,r,value)

The particular output quantity of interest for the solid cylinder at radius *r* is calculated in this subroutine based on the value of the variable *tft*. Cases of *emt* = 2 and 3 are applicable. The calculated quantity is returned by the variable *value*.

Subroutine MINV(c,cinv,work,n,iflag)

With this subroutine, complex system matrices are inverted for all permissible values of the variable *emt*. Matrix *c* is passed to *MINV* for inversion and *cinv* is the inverted matrix that is returned from the subroutine.

SUBPROGRAM c1.f

Subroutine C1A_POT(r,k,Om,n)

The displacement potentials for the first (inner) cylinder, evaluated at radius $r = a$, are calculated in this subroutine. This subroutine is also called when the calculation radius falls within the inner cylinder (*emt* = 1, 3, and 4); in this case, *r* is passed to the subroutine. The following variables are calculated and returned in Common Block /CYLINDER1/:

```
C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
        complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
        complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
        complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
        complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
        complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
        complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
```

Subroutine C1B_POT(r,k,Om,n)

The displacement potentials for cylinder 2, evaluated at radius $r = b$, are calculated in this subroutine. The following variables are calculated and returned in Common Block /CYLINDER1/:

```
C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
        complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
        complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
        complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
        complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
        complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
        complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
```

Subroutine SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl,bo_c1,Om,ro,co,smrc1)

The system matrix for *emt* = 3 is calculated in this subroutine and is returned as *smrc1*.

Subroutine ABC_RC1_INVERT(n,exctype,smrc1,b)

Undetermined coefficients are solved for through the inversion of matrix *smrc1* (*emt* = 3). The subroutine *MINV* is called from *ABC_RC1_INVERT* to perform the complex matrix inversion. The following variables are calculated and returned in Common Blocks /ROD/, /OFLUID/, and /CYLINDER1/:

```
C DEFINITION FOR COMMON BLOCK /ROD/
        complex*16 A1_rod,B1_rod,C1_rod

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        complex*16 M_OF

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
        complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
```

Subroutine OUTPUT_RC1(tft,n,k,r,value)

When the desired radius for output falls within the first, or inner, cylinder, this subroutine is used. Cases of *emt* = 1, 3, and 4 are applicable. The calculated quantity is returned by the variable *value*.

SUBPROGRAM c2.f

Subroutine C2B_POT(r,k,Om,n)

The displacement potentials for the second (outer) cylinder, evaluated at radius $r = b$, are calculated with this subroutine. This subroutine is also called when the calculation radius falls within the outer cylinder (*emt* = 4); in this case, r is passed to the subroutine. The following variables are calculated and returned in Common Block /CYLINDER2/:

```
C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

        complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
        complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
        complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
        complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
        complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
        complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
```

Subroutine C2C_POT(r,k,Om,n)

The displacement potentials for cylinder 2, evaluated at radius $r = c$, are calculated with this subroutine. The following variables are calculated and returned in Common Block /CYLINDER2/:

```
C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

        complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
        complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
        complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
        complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
        complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
        complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2
```

Subroutine SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2,Om,ro,co,ri,ci,smrc2)

The system matrix *smrc2* for *emt* = 4 is calculated and returned with this subroutine.

Subroutine ABC_RC2_INVERT(n,exctype,smrc2,a,c)

Undetermined coefficients are solved for through the inversion of matrix *smrc2* (*emt* = 4). The subroutine *MINV* is called from *ABC_RC2_INVERT* to perform the complex matrix inversion. The following variables are returned in Common Blocks /OFLUID/, /CYLINDER1/, /CYLINDER2/, and /IFLUID/:

```
C DEFINITIONS FOR COMMON BLOCK /OFLUID/
          complex*16 M_OF

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
          complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
          complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

          complex*16 D_IF
```

Subroutine OUTPUT_RC2(tft,n,k,r,value)

When the desired radius for output falls within the second or outer cylinder, this subroutine is used. Cases of *emt* = 3 and 4 are applicable. The calculated quantity is returned by the variable *value*.

SUBPROGRAM fluids.f

Subroutine IFL_POT(n,r,k,Om,ci)

Inner fluid displacement potentials are calculated with this subroutine. This subroutine is called once at $r = a$ in order to facilitate the solution of the undetermined coefficients. It is called a second time for *emt* = 1 and 4 when the desired calculation radius corresponds to the region of the inner fluid. The following variables are calculated and returned in Common Block /IFLUID/:

```
C DEFINITIONS FOR COMMON BLOCK /IFLUID/

          complex*16 IFSC,d1_IFSC
```

Subroutine OFL_POT(n,r,k,Om,co)

Outer fluid displacement potentials are calculated with this subroutine. This subroutine is called once at $r = b$ for *emt* = 1, 2, and 3 and once at $r = c$ for *emt* = 4. It is called a second time for all model cases where the desired calculation radius corresponds to the region of the outer fluid. The following variables are returned in the Common Block /OFLUID/:

```
C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        complex*16 OFSC,d1_OFSC
```

Subroutine OUTPUT_IF(tft,n,k,r,value,Om,ri)

If the radius at which output is desired corresponds to the region occupied by the inner fluid, this subroutine is used to calculate either the pressure or velocity transfer function based on the value of *tft*. The calculated quantity is returned by the variable *value*.

Subroutine OUTPUT_OF(tft,n,k,r,value,Om,ro)

If the radius at which output is desired corresponds to the region occupied by the outer fluid, this subroutine is used to calculate either the pressure or velocity transfer function based on the value of *tft*. The calculated quantity is returned by the variable *value*.

SUBPROGRAM smc1.f

Subroutine SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,Om,ro,co,ri,ci,smc1)

The system matrix *smc1* for *emt* = 1 is calculated and returned with this subroutine.

Subroutine ABC_C1_INVERT(n,exctype,smc1,a,b)

Undetermined coefficients are solved for through the inversion of matrix *smrc1* (*emt* = 1). The subroutine *MINV* is called from *ABC_C1_INVERT* to perform the complex matrix inversion. The following variables are calculated and returned in Common Blocks

/OFLUID/, /CYLINDER1/, and /IFLUID:

```
C DEFINITIONS FOR COMMON BLOCK /OFLUID/

          complex*16 M_OF

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
          complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

          complex*16 D_IF
```

FLOW CHARTS

INNER FLUID/CYLINDER 1/OUTER FLUID

This case, *emt* = 1, involves three media: the inner fluid, cylinder 1, and the outer fluid. Flow chart 1 in figure 3 displays the necessary execution sequence for this model. Subroutines C1A_POT and C1B_POT calculate the Bessel functions needed for the cylinder displacement potentials at the inner and outer radii of the cylinder, respectively. OFL_POT and IFL_POT calculate the Bessel functions needed for the displacement potentials of the outer and inner fluids, respectively. The results of these four subroutine calls are passed to subroutine SYS_MATRIX_C1 via common blocks, and then the components of the system matrix, *smc*1, are calculated. The undetermined coefficients are solved for after system matrix *smc*1 is inverted in subroutine ABC_C1_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the three media mentioned above. The output calculations are partitioned by the three *if* statements, and the output quantity is calculated based on the value of the radius r_1 specified for the output calculation. If r_1 is less than or equal to the inner radius of the cylinder, the IFL_POT subroutine is used to calculate the Bessel functions needed for the inner fluid displacement potential and then OUTPUT_IF is used to calculate the desired output quantity according to the value of variable *tft*. If r_1 is less than or equal to the outer radius of the cylinder and greater than the inner radius, then C1A_POT is used to calculate the Bessel functions needed for the displacement potentials and OUTPUT_RC1 is used to calculate the desired output quantity according to the value of the variable *tft*. When r_1 is greater than the outer radius of the cylinder, subroutine OFL_POT is called, and the Bessel functions needed for the outer fluid displacement potential at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable *tft*.

The subroutines listed in appendix D have been modified to account for the case of circumferential excitation. Substitution of the appendix D subroutines into subprograms *rf.f*, *smc1.f*, *c1.f*, and *c2.f* is required in order to calculate the response from circumferential excitation P_θ.

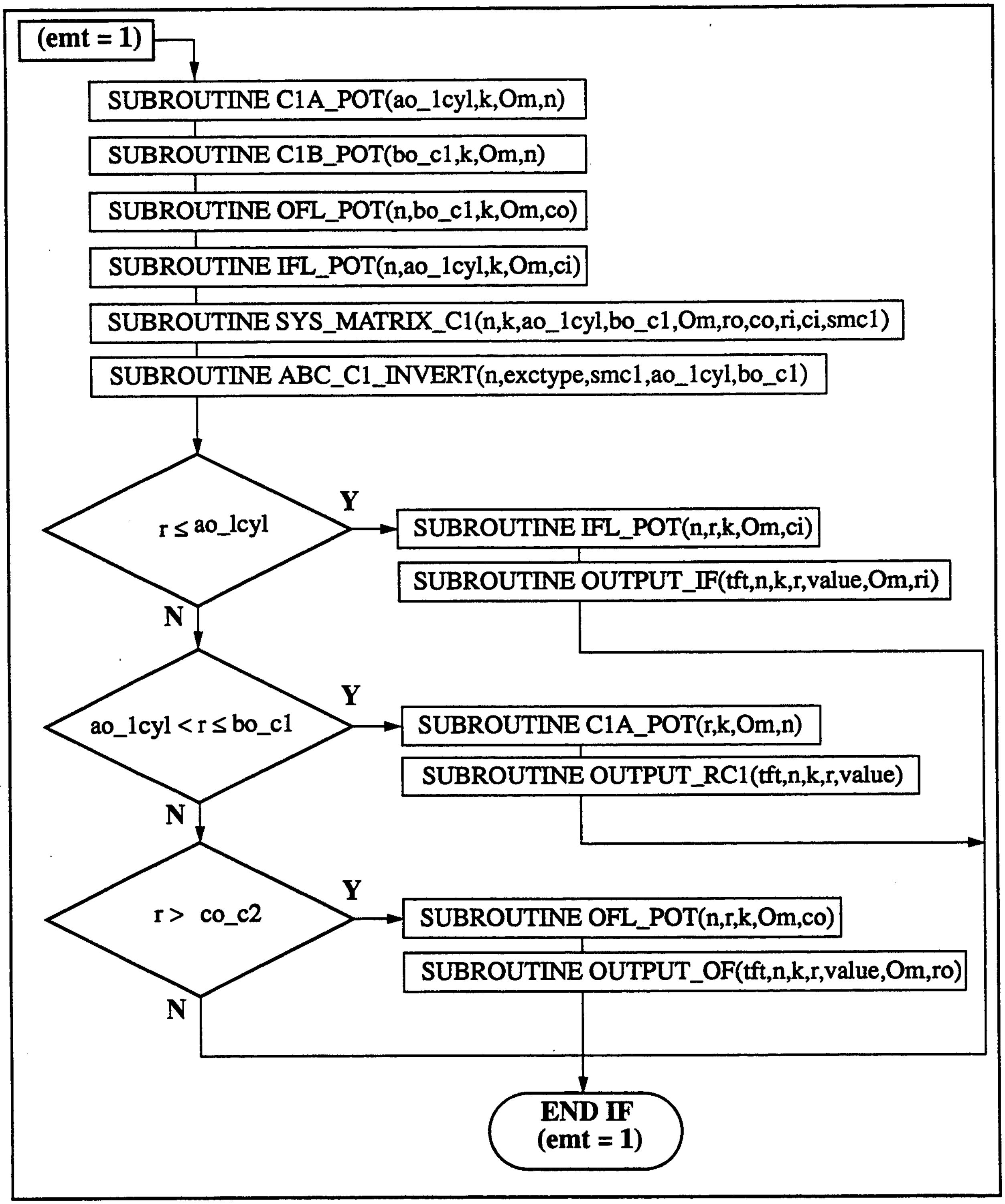


Figure 3. Flow Chart 1—Inner Fluid, Cylinder 1, and Outer Fluid for *emt* = 1

SOLID CYLINDER/OUTER FLUID

This case, *emt* = 2, involves two media: the solid cylinder and the outer fluid. In figure 4, flow chart 2 displays the necessary execution sequence for this model. Subroutine ROD_POT calculates the Bessel functions needed for the solid cylinder displacement potentials evaluated at the outer radius of the cylinder. Subroutine OFL_POT calculates the Bessel functions needed for the outer fluid displacement potential. The results of these two subroutine calls are passed to subroutine SYS_MATRIX_ROD via common blocks, and the components of the system matrix *sm* are calculated. The undetermined coefficients are solved for after system matrix *sm* is inverted in subroutine ABC_ROD_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the two media mentioned above. The output calculations are partitioned by the two *if* statements, and the output quantity is calculated based on the value of the radius (r_1) specified for output calculation. If r_1 is less than the outer radius of the cylinder, then ROD_POT is used to calculate the Bessel functions needed for the displacement potentials and OUTPUT is used to calculate the desired output quantity according to the value of the variable *tft*. When r_1 is greater than the outer radius of the cylinder, subroutine OFL_POT is called and the Bessel functions needed for the outer fluid displacement potential at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable *tft*.

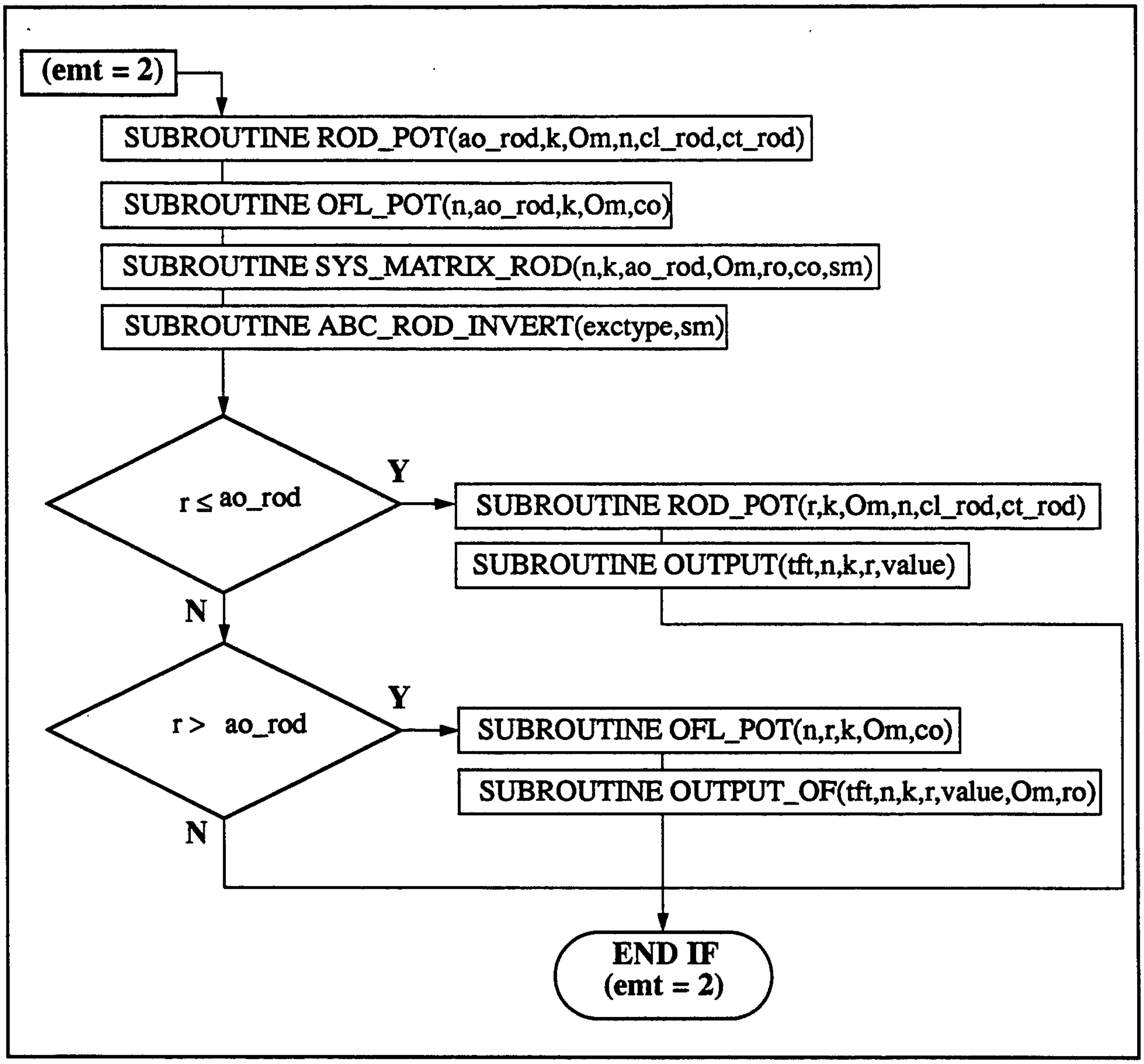


Figure 4. Flow Chart 2—Solid Cylinder and Outer Fluid for *emt* = 2

SOLID CYLINDER/CYLINDER 2/OUTER FLUID

This case, *emt* = 3, involves three media: the solid cylinder, cylinder 2, and the outer fluid. In figure 5, flow chart 3 displays the necessary execution sequence for this model. Subroutines C1A_POT and C1B_POT calculate the Bessel functions needed for the displacement potentials of cylinder 2 at the inner and outer radii of the cylinder, respectively. Subroutine ROD_POT calculates the Bessel functions needed for the solid cylinder displacement potentials at $r = a$. OFL_POT calculates the Bessel functions needed for the outer fluid displacement potential at $r = b$. The results of these four subroutine calls are passed to subroutine SYS_MATRIX_RC1 via common blocks, and then the components of the system matrix, *smrc*1, are calculated. The undetermined coefficients are solved for after system matrix *smrc*1 is inverted in subroutine ABC_RC1_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the three media mentioned above. The output calculations are partitioned by the three *if* statements, and the output quantity is calculated based on the value of the radius (r_1) specified for output calculation. If r_1 is less than or equal to the outer radius of the solid cylinder, the ROD_POT subroutine is used to calculate the Bessel functions needed for the displacement potentials at radius r_1 and then OUTPUT is used to calculate the desired output quantity according to the value of variable *tft*. If r_1 is less than or equal to the outer radius of cylinder 2 and greater than the inner radius, then C1A_POT is used to calculate the Bessel functions needed for the displacement potentials and OUTPUT_RC1 is used to calculate the desired output quantity according to the value of the variable *tft*. When r_1 is greater than the outer radius of cylinder 2, subroutine OFL_POT is called and the Bessel functions needed for the outer fluid displacement potential at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable *tft*.

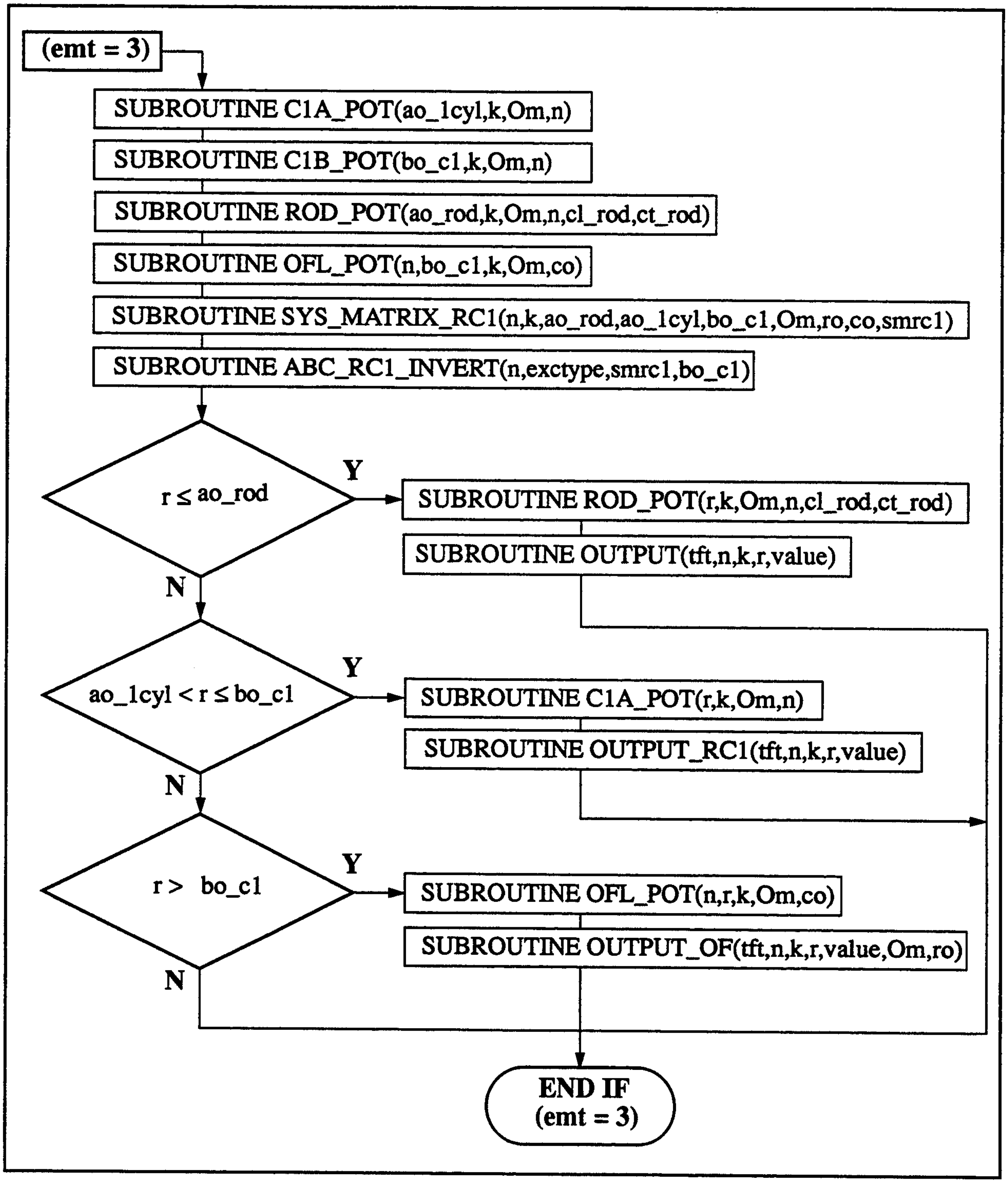


Figure 5. Flow Chart 3—Solid Cylinder, Cylinder 2, and Outer Fluid for *emt* = 3

INNER FLUID/CYLINDER 1/CYLINDER 2/OUTER FLUID

This case, *emt* = 4, involves four media: the inner fluid, cylinder 1, cylinder 2, and the outer fluid. In figure 6, flow chart 4 displays the necessary execution sequence for this model. Subroutines C1A_POT and C1B_POT calculate the Bessel functions needed for the displacement potentials of cylinder 1 at the inner and outer radii of the cylinder, respectively. Subroutines C2B_POT and C2C_POT calculate the Bessel functions needed for the displacement potentials of cylinder 2 at $r = b$ and $r = c$, respectively. OFL_POT calculates the Bessel functions needed for the displacement potential of the outer fluid at $r = c$. The results of these six subroutine calls are passed to subroutine SYS_MATRIX_RC2 via common blocks, and then the components of the system matrix, *smrc*2, are calculated. The undetermined coefficients are solved for after system matrix *smrc*2 is inverted in subroutine ABC_RC2_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the four media mentioned above. The output calculation is partitioned by the four *if* statements, and the output quantity is calculated based on the value of the radius (r_1) specified for output calculation. If r_1 is less than or equal to the inner radius of cylinder 1, IFL_POT is called to evaluate the Bessel functions needed for the fluid displacement potential. Subroutine OUTPUT_IF is used to calculate the desired output quantity according to the value of the variable *tft*. If r_1 is less than or equal to the outer radius of cylinder 1 and greater than the inner radius, subroutine C1A_POT is used to calculate the Bessel functions needed for the displacement potentials of cylinder 1 at radius r_1 and then subroutine OUTPUT is used to calculate the desired output quantity according to the value of the variable *tft*. If r_1 is less than or equal to the outer radius of cylinder 2 and greater than the inner radius, then C2B_POT is used to calculate the Bessel functions needed for the displacement potentials at radius r_1 and OUTPUT_RC2 is used to calculate the desired output quantity according to the value of the variable *tft*. When r_1 is greater than the outer radius of cylinder 2, subroutine OFL_POT is called and the Bessel functions needed for the displacement potential of the outer fluid at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable *tft*.

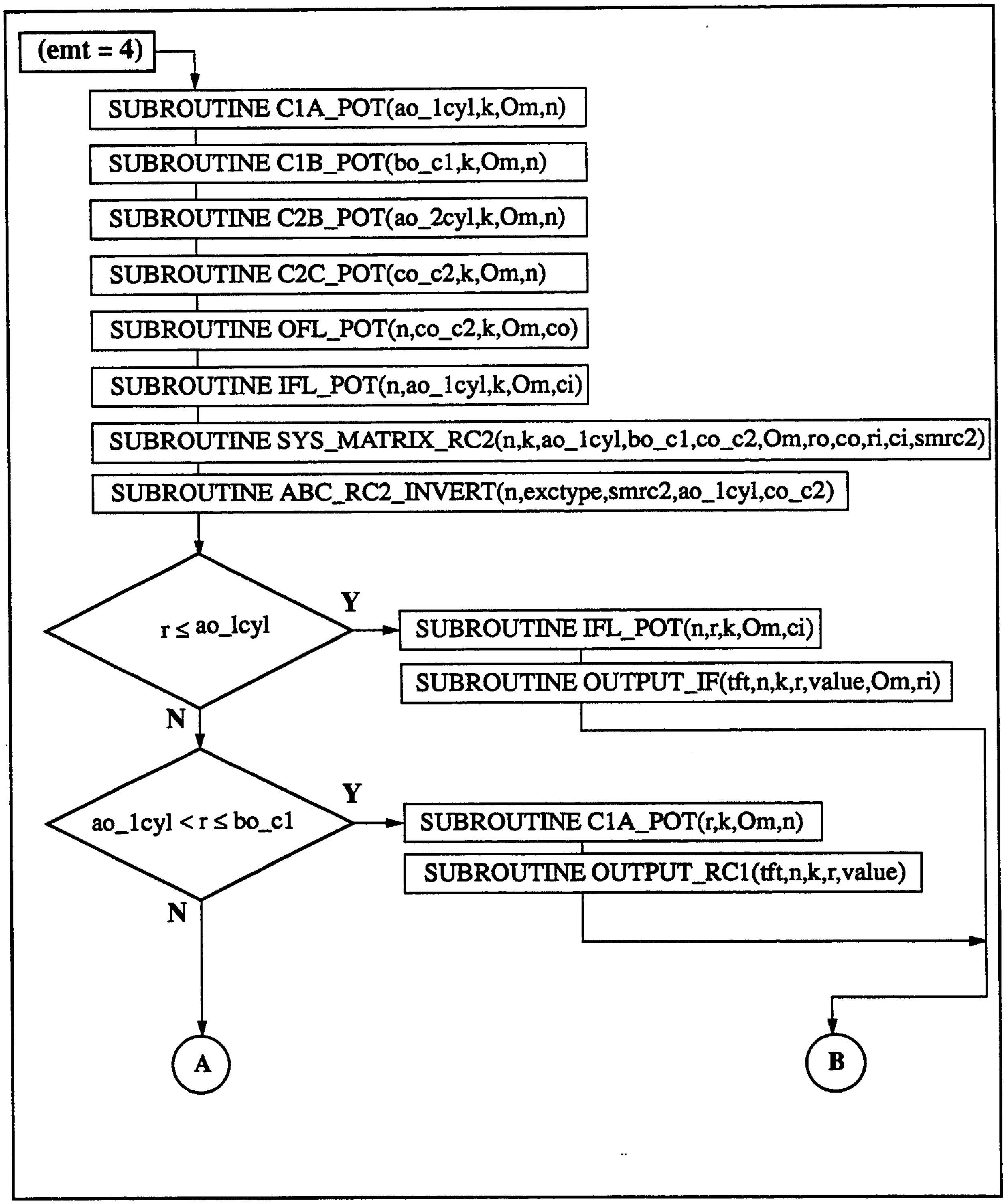


Figure 6. Flow Chart 4—Inner Fluid, Cylinder 1, Cylinder 2, and Outer Fluid for *emt* = 4

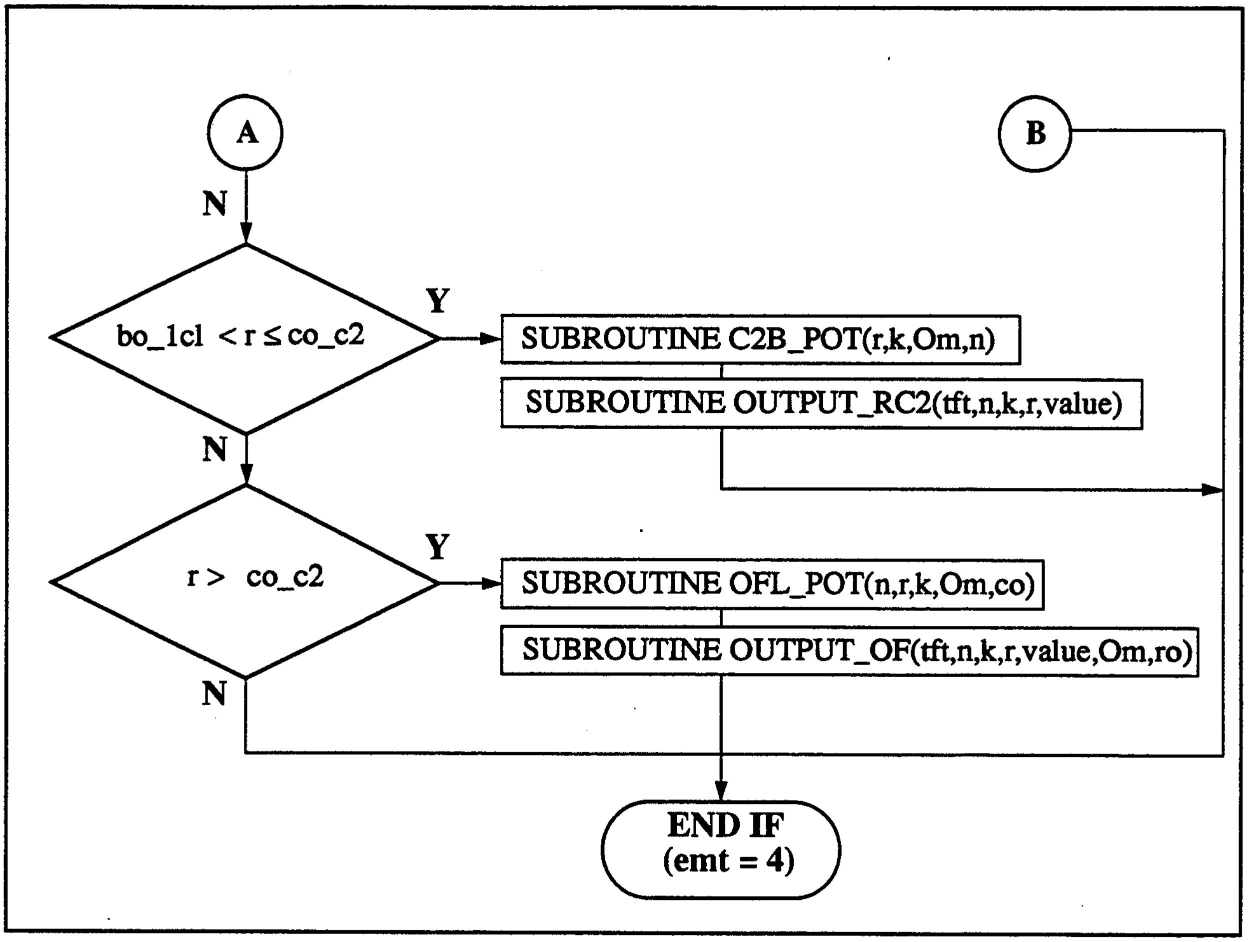


Figure 6. Flow Chart 4—Inner Fluid, Cylinder 1, Cylinder 2, and Outer Fluid for *emt* = 4 (Cont'd)

SUBPROGRAM LISTINGS

LISTING FOR mr2cf.f

```
C TEST PROGRAM “mr2cf.f FOR TESTING THE TWO CYLINDER INNER/OUTER FLUID
C SIMULATION 5/16/95
C TEST PROGRAM VARIABLES
C         character*1 msg
          complex*16 a,z,Jn,In,Yn,Kn,mains
          real*8 SI,bb
          integer FA,msg,f
C **********************************************************************
C VARIABLES NEEDED TO SIMULATE MAIN PROGRAM Xtota
          integer iptmax
          parameter (iptmax = 80000)
          integer n,tft,exctype,jk,emt
          real*8 r,ao_rod,mu_rod,r_rod
          real*8 k,Om,co,ro,zeta_rod
          complex*16 g(iptmax),E_rod
          real*8 ri,ci
          real*8 E_1cyl,zeta_1cyl,mu_1cyl,ao_1cyl,r_1cyl,h_1cyl
          real*8 E_2cyl,zeta_2cyl,mu_2cyl,ao_2cyl,r_2cyl,h_2cyl
C **********************************************************************
C NEW ADDITIONS TO MAIN PROGRAM
          integer gamma,fac,iflag,size
          real*8 psi
          complex*16 cl_rod,ct_rod,sm(4,4)
          complex*16 work(4,8),sminv(4,4)
          complex*16 value
          complex*16 Ec_rod
          double complex cbessj,cbessi,cbessy,cbessk,cbessh1,cbessh2
          double complex d1cbessj,d2cbessj
          double complex d1cbessy,d2cbessy
          double complex d1cbessk,d2cbessk
          double complex d1cbessi,d2cbessi
          double complex d1cbessh1,d1cbessh2
```

```
C ADDITIONS NEEDED FOR THE FIRST CYLINDER BEYOND THE ROD/FLUID
        real*8 bo_c1
        complex*16 smrc1(10,10),smrc1inv(10,10),workrc1(10,20)
        complex*16 Ec_c1
C  ADDITIONS NEEDED FOR THE SECOND CYLINDER BEYOND THE ROD/ CYLINDER
        real*8 co_c2
        complex*16 smrc2(13,13),smrc2inv(13,13),workrc2(13,26)
        complex*16 Ec_c2
C  ADDITIONS NEEDED FOR CYLINDER WITH INNER AND OUTER FLUIDS ONLY
        complex*16 smc1(7,7),smc1inv(7,7),workc1(7,14)
C ********************************************************************
C DEFINITION FOR COMMON BLOCK /ROD/
        complex*16 SP_rod,d1_SP_rod,d2_SP_rod
        complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
        complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
        complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
        common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
   1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
   1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
C ********************************************************************
C ********************************************************************
C DEFINITIONS FOR COMMON BLOCK /OFLUID/
        complex*16 OFSC,d1_OFSC,M_OF
        common /OFLUID/ OFSC,d1_OFSC,M_OF
C ********************************************************************
```

```
C **********************************************************************

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1          SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1          SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1          SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1          VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1          VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1          VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1          VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1          VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1          VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1          VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1          VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1          lame_c1,shear_c1,cl_c1,ct_c1,
     1          A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C **********************************************************************
```

```
C “CYLINDER 2 VARIABLES NEEDED TO EXPAND BEYOND ROD/CYLINDER CASE **

C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
     1          SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
     1          SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
     1          SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
     1          VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
     1          VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
     1          VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
     1          VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
     1          VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
     1          VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
     1          VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
     1          VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
     1          lame_c2,shear_c2,cl_c2,ct_c2,
     1          A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C ***********************************************************************

C ***********************************************************************
```

```
C DEFINITIONS FOR COMMON BLOCK /IFLUID/

        complex*16 IFSC,d1_IFSC,D_IF

        common /IFLUID/ IFSC,d1_IFSC,D_IF


C ***********************************************************************


C "ROD CASE" VARIABLES NEEDED TO SIMULATE MAIN PROGRAM

        E_rod    =   7.20D2
        mu_rod   =   0.17D0
        r_rod    = 2600.0D0
        zeta_rod =   0.0D0

C "ROD CASE" MAIN PROGRAM CALCULATION SECTION ADDITIONS *****

        Ec_rod    = dcmplx(1.0D0,zeta_rod)*E_rod
        lame_rod  = Ec_rod*mu_rod/((1.0D0+mu_rod)*(1.0D0-2.0D0*mu_rod))
        shear_rod = Ec_rod/(2.0D0*(1.0D0 + mu_rod))
        cl_rod    = zsqrt((lame_rod + 2.0D0* shear_rod)/r_rod)
        ct_rod    = zsqrt(shear_rod / r_rod)

C ***********************************************************************

C "ROD/CYLINDER" VARIABLES NEEDED TO SIMULATE MAIN PROGRAM

        E_1cyl    =   7.1D10
        zeta_1cyl =   0.0D0
        mu_1cyl   =   0.33D0
        ao_1cyl   =  125.0D-6
        r_1cyl    = 2700.0D0
        h_1cyl    =  375.0D-6


C "ROD/CYL1 CASE" MAIN PROGRAM CALCULATION SECTION ADDITIONS **

        Ec_c1    = dcmplx(1.0D0,zeta_1cyl)*E_1cyl
        lame_c1  = Ec_c1*mu_1cyl/((1.0D0+mu_1cyl)*(1.0D0-2.0D0*mu_1cyl))
        shear_c1 = Ec_c1/(2.0D0*(1.0D0+mu_1cyl))
        cl_c1    = zsqrt((lame_c1 + 2.0D0*shear_c1)/r_1cyl)
        ct_c1    = zsqrt(shear_c1/r_1cyl)
        bo_c1    = ao_1cyl + h_1cyl
```

```
C ***********************************************************************

C ***********************************************************************

C "TWO CYLINDER" VARIABLES NEEDED TO SIMULATE MAIN PROGRAM

        E_2cyl    =   7.1D10
        zeta_2cyl =   .0D0
        mu_2cyl   =   .33D0
        ao_2cyl   =   400.0D-6
        r_2cyl    = 2700.0D0
        h_2cyl    =   100.0D-6


C "ROD/CYL1 CASE" MAIN PROGRAM CALCULATION SECTION ADDITIONS *

        Ec_c2    = dcmplx(1.0D0,zeta_2cyl)*E_2cyl
        lame_c2  = Ec_c2*mu_2cyl/((1.0D0+mu_2cyl)*(1.0D0-2.0D0*mu_2cyl))
        shear_c2 = Ec_c2/(2.0D0*(1.0D0+mu_2cyl))
        cl_c2    = zsqrt((lame_c2 + 2.0D0*shear_c2)/r_2cyl)
        ct_c2    = zsqrt(shear_c2/r_2cyl)
        co_c2    = ao_2cyl + h_2cyl

C ***********************************************************************



C  RADIAL STRESS/(Pr or Px)                            tft = 0
C  LONGITUDINAL STRESS                                 tft = 1
C  AXIAL  DISPLACEMENT                                 tft = 2
C  THETA  DISPLACEMENT                                 tft = 3
C  RADIAL DISPLACEMENT                                 tft = 4
C  LONGITUDINAL STRAIN  e11/(Pr or Px)                 tft = 5
C  THETA  STRAIN        ett/(Pr or Px)                 tft = 6
C  RADIAL STRAIN        err/(Pr or Px)                 tft = 7
C                                                      tft = 8
C  OPTIC exx=0@k=0      ((dp/p)(r))/(Pr or Px)         tft = 9
C  OPTIC exx=const@k=0  ((dp/p)(r))/(Pr or Px)         tft = 10

C RADIAL EXCITATION       exctype = 1
C AXIAL  EXCITATION       exctype = 0

C INNER FLUID/CYLINDER/OUTER FLUID                     emt = 1
C ROD AND OUTER FLUID                                  emt = 2
C ROD/CYLINDER/FLUID                                   emt = 3
C INNER FLUID/TWO CYLINDERS/OUTER FLUID                emt = 4
```

```
      k      = 0.1D0
      Om     = 6000.0D0
      ao_rod = 125.0D-6
      r      = 150.0D-6
      n      = 0
      tft    = 5
      exctype = 1
      emt    = 3
      jk     = 1

      co     = 1500.0D0
      ro     = 1000.0D0

      ci              = 150.0D0
      ri              = 1.0D0

      g(jk)  = 3.0D0
      size   = 3
      iflag  = 0

      mains = zsqrt(k**2 - (Om**2/cl_rod**2))*ao_rod

      msg = 0

      do while (msg .ne. 1)

      print *,'Enter tft,k,f,emt: (where emt=1 1CYLINDER,
     1 emt=2 ROD,  emt=3 ROD&CYL,  emt=4 2CYLINDERS,
     1using mr2cf.f)'

      read(*,20)tft,k,f,emt
 20   format(i2,f10.4,i6,i1)

      Om = 2.0D0*3.14D0*f

        write(*,10)n,r,ao_1cyl,bo_c1,co_c2,tft
 10     format(/,'MAIN PROGRAM  n = ',i2,'   r = ',E15.7,
     1' ao_1cyl = ',E15.7,' bo_c1 = ',E15.7,' co_c2 = ',E15.7,/
     1,'tft = ',i2)
```

```
C INNER FLUID/CYLINDER 1/OUTER FLUID (emt=1) **********************
        if (emt .eq. 1) then


        CALL C1A_POT(ao_1cyl,k,Om,n)
        CALL C1B_POT(bo_c1,k,Om,n)
        CALL OFL_POT(n,bo_c1,k,Om,co)
        CALL IFL_POT(n,ao_1cyl,k,Om,ci)
        CALL SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,
     1Om,ro,co,ri,ci,smc1)
        CALL ABC_C1_INVERT(n,exctype,smc1,ao_1cyl,bo_c1)

          if (r .le. ao_1cyl) then

C       PRESSURE AND VELOCITY TRANSFER FUNCTIONS (INNER FLUID)
                CALL IFL_POT(n,r,k,Om,ci)
                CALL OUTPUT_IF(tft,n,k,r,value,Om,ri)

          elseif (r .le. bo_c1 .and. r .gt. ao_1cyl) then

                CALL C1A_POT(r,k,Om,n)
                CALL OUTPUT_RC1(tft,n,k,r,value)

C       PRESSURE AND VELOCITY TRANSFER FUNCTIONS (OUTER FLUID)
          elseif (r .gt. co_c2) then

                CALL OFL_POT(n,r,k,Om,co)

        write(*,400)d1_OFSC,OFSC,M_OF
400     format(/,'d1_OFSC = ',2E15.7,' OFSC = ',2E15.7,' M_OF = ',
     12E15.7,/)

                CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)

        endif

        if(tft .gt. 1)then
          value = value*(1.0D-6)
        endif

        g(jk) = value
C ***********************************************************************
```

```
C SOLID CYLINDER/OUTER FLUID (emt=2) ********************************
        elseif (emt .eq. 2) then

        CALL ROD_POT(ao_rod,k,Om,n,cl_rod,ct_rod)
        CALL OFL_POT(n,ao_rod,k,Om,co)
        CALL SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)
        CALL ABC_ROD_INVERT(exctype,sm)

          if (r .le. ao_rod) then

              CALL ROD_POT(r,k,Om,n,cl_rod,ct_rod)
              CALL OUTPUT(tft,n,k,r,value)

C       PRESSURE AND VELOCITY TRANSFER FUNCTIONS
        elseif (r .gt. ao_rod) then

              CALL OFL_POT(n,r,k,Om,co)
              write(*,400)d1_OFSC,OFSC,M_OF
              CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)

        endif

        if(tft .gt. 1)then
          value = value*(1.0D-6)
        endif

        g(jk) = value
C ***********************************************************************
```

```
C SOLID CYLINDER/CYLINDER 1/ OUTER FLUID (emt=3) ********************
        elseif (emt .eq. 3)then

        CALL C1A_POT(ao_1cyl,k,Om,n)
        CALL C1B_POT(bo_c1,k,Om,n)
        CALL ROD_POT(ao_rod,k,Om,n,cl_rod,ct_rod)
        CALL OFL_POT(n,bo_c1,k,Om,co)
        CALL SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl,bo_c1,Om,ro,co
     1,smrc1)
        CALL ABC_RC1_INVERT(n,exctype,smrc1,bo_c1)

        if (r .le. ao_rod) then

              CALL ROD_POT(r,k,Om,n,cl_rod,ct_rod)
              CALL OUTPUT(tft,n,k,r,value)

        elseif (r .le. bo_c1 .and. r .gt. ao_rod) then

              CALL C1A_POT(r,k,Om,n)
              CALL OUTPUT_RC1(tft,n,k,r,value)

C       PRESSURE AND VELOCITY TRANSFER FUNCTIONS
        elseif (r .gt. bo_c1) then

              CALL OFL_POT(n,r,k,Om,co)
        write(*,400)d1_OFSC,OFSC,M_OF

              CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)

        endif

        if(tft .gt. 1)then
          value = value*(1.0D-6)
        endif

        g(jk) = value

C ******************************************************************
```

```
C INNER FLUID/CYLINDER 1/CYLINDER 2/OUTER FLUID ******************
      elseif (emt .eq. 4)then

            CALL C1A_POT(ao_1cyl,k,Om,n)
            CALL C1B_POT(bo_c1,k,Om,n)
            CALL C2B_POT(ao_2cyl,k,Om,n)
            CALL C2C_POT(co_c2,k,Om,n)
            CALL IFL_POT(n,ao_1cyl,k,Om,ci)

      write(*,410)d1_IFSC,IFSC,D_IF
410   format(/,'d1_IFSC = ',2E15.7,' IFSC = ',2E15.7,' D_IF = ',
  12E15.7,/)
            CALL OFL_POT(n,co_c2,k,Om,co)
            CALL SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2,Om,ro,co
  1,ri,ci,smrc2)
            CALL ABC_RC2_INVERT(n,exctype,smrc2,ao_1cyl,co_c2)

      write(*,400)d1_OFSC,OFSC,M_OF

        if (r .le. ao_1cyl) then

C       PRESSURE AND VELOCITY TRANSFER FUNCTIONS (INNER FLUID)
            CALL IFL_POT(n,r,k,Om,ci)
            CALL OUTPUT_IF(tft,n,k,r,value,Om,ri)

        elseif (r .le. bo_c1 .and. r .gt. ao_1cyl) then

            CALL C1A_POT(r,k,Om,n)
            CALL OUTPUT_RC1(tft,n,k,r,value)

        elseif (r .le. co_c2 .and. r .gt. bo_c1) then

            CALL C2B_POT(r,k,Om,n)
            CALL OUTPUT_RC2(tft,n,k,r,value)

C      PRESSURE AND VELOCITY TRANSFER FUNCTIONS (OUTER FLUID)
        elseif (r .gt. co_c2) then

            CALL OFL_POT(n,r,k,Om,co)

      write(*,400)d1_OFSC,OFSC,M_OF

            CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)

        endif

```

```
      if(tft .gt. 1)then
        value = value*(1.0D-6)
      endif

      g(jk) = value

      endif

      g(jk) = 10.0D0*log10(cdabs((value)**2))

C ********************************************************************

      write (*,51)value
51      format('SUB OUTPUT THE ANSWER IS value = '2e15.7)
      write (*,60)g(jk)
60      format('SUB OUTPUT THE ANSWER IS g(jk) = '2e15.7)
      write (*,70)k,Om
70      format('SUB OUTPUT                k = ',e15.7,
     1'  Om = ',f7.2,/)




      print *,'Type 1 if you wish to quit'
      read(*,4) msg
 4    format(i1)

      end do
      stop
      end
```

LISTING FOR cbessl.f

cbessl.f Thu Oct 19 16:47:22 1995

```
      function gamma(n)
      integer i,n,sum,gamma
        if(n .eq. 0 .or. n .eq. 1) then
          gamma = 1
        else
          sum=n
          do 10, i=1, n-1
          sum = sum*(n-i)
 10       continue
          gamma = sum
        endif
      return
      end




      function fac(n)
      integer n,sum,fac,i
        if(n .eq. 0 .or. n .eq. 1) then
          fac = 1
        else
          sum=n
          do 10, i=1, n-1
             sum = sum*(n-i)
 10       continue
        fac = sum
        endif
      return
      end
```

```
      function psi(n)
      integer n,na,i
      real*8 sum,psi,euler
      parameter (euler=.5772156649015328606)
      sum = 0.0
      na = iabs(n)
        if (na .eq. 1) then
          psi = - euler
        else
          do 10, i=1, na-1
          sum = sum + (1.0/i)
10     continue
          psi = -euler + sum
        endif
      return
      end

      double complex function cbessj(n,a,r)

       integer n,limit,j,k,na,fac
      real*8 r,pi,zm
      complex*16 a,z,sum,total,Am,Bm
      double complex cbessj

      real*8 fn2
      complex*16 a1,a2,a3,a4,b1,b2,b3,b4,ez,ez2

      parameter (pi=3.14159265358979323 84D0)

      z = a*r
      na = iabs(n)
      zm = cdabs(z)

      fn2 = 4.0D0*n**2
      ez  = 8.0D0*z
      ez2 = (8.0D0*z)**2

        if (zm .le. 3.0) then

          limit = 10
          total = (0.0D0,0.0D0)
```

```
C Abramowitz and Stegum Equation 9.1.10

              do 10, k=0, limit
              j = na + k
              sum = ((-0.25D0*z**2)**k)/(fac(k)*fac(j))
              total = total + sum
 10           continue
              cbessj = ((z/2.0D0)**na)*total

              if (n .lt. 0) then
                  cbessj = ((-1.0D0)**na)*cbessj


              end if

            else

C Large Argument Calculation  Korn and Korn  21.8-44 and 45


              a1 =    (fn2 - 1.0D0)*(fn2 - 9.0D0)/(2.0D0*ez2)

              a2 = a1*(fn2 - 25.0D0)*(fn2 - 49.0D0)/(ez2*12.0D0)

              a3 = a2*(fn2 - 81.0D0)*(fn2 - 121.0D0)/(ez2*30.0D0)

              a4 = a3*(fn2 - 169.0D0)*(fn2 - 225.0D0)/(ez2*56.0D0)


              Am = 1.0D0 - a1 + a2 - a3 + a4

          b1 =    (fn2 - 1.0D0)/ez

          b2 = b1*(fn2 - 9.0D0)*(fn2 - 25.0D0)/(ez2*6.0D0)

          b3 = b2*(fn2 - 49.0D0)*(fn2 - 81.0D0)/(ez2*20.0D0)

          b4 = b3*(fn2 - 121.0D0)*(fn2 - 169.0D0)/(ez2*42.0D0)


              Bm = b1 - b2 + b3 - b4

          cbessj = cdsqrt(2.0D0/(pi*z))*(Am*cdcos(z - n*pi/
     1       2.0D0 - pi/4.0D0) - Bm*cdsin(z - n*pi/2.0D0 - pi/
     1       4.0D0))
```

```
      endif



      return
      end




      double complex function cbessi(n,a,r)

      integer n,limit,j,k,na,fac
      real*8 r
      complex*16 a,z,sum,total
      double complex cbessi

      z = a*r
      na=iabs(n)
      limit = 10
      total = (0.0,0.0)

      do 10, k=0, limit
         j = na + k
         sum = ((.25*z**2)**k)/(fac(k)*fac(j))
         total = total + sum
10    continue
      cbessi = ((z/2)**na)*total


      return
      end
```

```
      double complex function cbessy(n,a,r)

      integer n,na,limit,j,k,l,m,fac
      real*8 r,pi,psi,zm
      complex*16 a,z,z2,part1,part2,part3,sum1,sum2
      complex*16 total1,total2,Am,Bm
      double complex cbessj,cbessy

      parameter (pi=3.1415926535897932384)

      z = a*r
      z2 = .5*z
      na = iabs(n)
      zm = cdabs(z)

      IF (zm .le. 3.0) THEN

C Abramowitz and Stegum Equation 9.1.11

      part1 = (0.,0.)
      part2 = (0.,0.)
      part3 = (0.,0.)
      total1 = (0.,0.)
      total2 = (0.,0.)
      limit = 10

      do 10, k = 0, na-1
         j = na-k-1
         sum1 = (fac(j)/fac(k))*(.25*z**2)**k
         total1 = total1 + sum1
 10   continue

      part1 = (-((.5*z)**-na)/pi)*total1

      part2 = (2.0/pi)*cdlog(z2)*cbessj(na,a,r)

      do 20, k = 0, limit
         j = k+1
         l = na+k+1
         m = na+k
         sum2 = (psi(j)+psi(l))*(-.25*z**2)**k/(fac(k)*fac(m))
         total2 = total2+sum2
 20   continue

      part3 = ((-(.5*z)**na)/pi)*total2

```

```
      cbessy = part1+part2+part3


      if (n .lt. 0) then
         cbessy = ((-1)**na)*cbessy
      end if


      ELSE

C Large Argument Calculation  Korn and Korn  21.8-44 & 45

      Am = 1.0 - (4.0*n**2 - 1.0)*(4*n**2 - 9.0)/(2.0*(8*z)**2)
     1 + (4.0*n**2-1.0)*(4.0*n**2-9.0)*(4.0*n**2-25.0)*
     1 (4.0*n**2-49.0)/(24.0*(8.0*z)**4)

      Bm = (4.0*n**2-1.0)/(8.0*z) - (4.0*n**2-1.0)*(4.0*n**2-9.0)*
     1 (4.0*n**2-25.0)/(6.0*(8.0*z)**3)

      cbessy = cdsqrt(2.0/(pi*z))*(Am*cdsin(z - n*pi/2.0 - pi/4.0) +
     1    Bm*cdcos(z - n*pi/2.0 - pi/4.0))

      ENDIF


      return
      end
```

```
      double complex function cbessk(n,a,r)

      integer n,na,limit,j,k,l,m,fac
      real*8 r,pi,psi
      complex*16 a,z,z2,part1,part2,part3,sum1,sum2,total1,total2
      double complex cbessi,cbessk

      parameter (pi=3.14159265358979323 84)

      z = a*r
      z2 = .5*z
      na = iabs(n)


      part1 = (0.,0.)
      part2 = (0.,0.)
      part3 = (0.,0.)
      total1 = (0.,0.)
      total2 = (0.,0.)
      limit = 10

      do 10, k = 0, na-1
         j = na-k-1
         sum1 = (fac(j)/fac(k))*(-.25*z**2)**k
         total1 = total1 + sum1
10    continue

      part1 = (((.5*z)**-na)/2.)*total1

      part2 = ((-1)**(na+1))*cdlog(z2)*cbessi(na,a,r)

      do 20, k = 0, limit
         j = k+1
         l = na+k+1
         m = na+k
         sum2 = (psi(j)+psi(l))*(.25*z**2)**k/(fac(k)*fac(m))
         total2 = total2+sum2
20    continue

      part3 = ((-1)**na)*((.5*z)**na)*.5*total2

      cbessk = part1+part2+part3

      return
      end

```

```
      double complex function d1cbessk(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessk,d1cbessk

      d1cbessk = -a*cbessk(n+1,a,r)+(n/r)*cbessk(n,a,r)

      return
      end




      double complex function d2cbessk(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex d2cbessk

      d2cbessk = (1.0,1.0)

      return
      end




      double complex function d1cbessj(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessj,d1cbessj

        d1cbessj = -a*cbessj(n+1,a,r)+(n/r)*cbessj(n,a,r)

      return
      end
```

```
      double complex function d2cbessj(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessj,d2cbessj


        d2cbessj = ((a**2)/4.0D0)*(cbessj(n-2,a,r) - 2.0D0*
     1    cbessj(n,a,r) + cbessj(n+2,a,r))


      return
      end




      double complex function d1cbessy(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessy,d1cbessy
      complex*16 Y2,Y3


      IF (n .eq. 0) THEN

        d1cbessy = - a*cbessy(1,a,r)

      ELSEIF (n .eq. 1) THEN

        Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
        d1cbessy = - a*Y2 + cbessy(1,a,r)/r

      ELSEIF (n .eq. 2) THEN

        Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
        Y3 = (8.0D0/((a*r)**2) - 1.0D0)*cbessy(1,a,r) -
     1      4.0D0*cbessy(0,a,r)/(a*r)
        d1cbessy = - a*Y3 +2.0D0*Y2/r
```

```
      ELSEIF (n .eq. -1) THEN

        d1cbessy = -a*cbessy(0,a,r) - cbessy(-1,a,r)/r

      ELSEIF (n .eq. -2) THEN

        Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
        d1cbessy = -a*cbessy(-1,a,r) -2.0D0*Y2/r


      ENDIF


      return
      end




      double complex function d2cbessy(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessy,d2cbessy
      complex*16 Y2,Y3,Y4



      IF (n .eq. 0) THEN

        Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
        d2cbessy = a**2*Y2 - a*cbessy(1,a,r)/r

      ELSEIF (n .eq. 1) THEN

        Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
        Y3 = (8.0D0/((a*r)**2) - 1.0D0)*cbessy(1,a,r) -
     1        4.0D0*cbessy(0,a,r)/(a*r)
        d2cbessy = a**2*Y3 - 3.0D0*a*Y2/r

      ELSEIF (n .eq. 2) THEN

        Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
        Y3 = (8.0D0/((a*r)**2) - 1.0D0)*cbessy(1,a,r) -
     1        4.0D0*cbessy(0,a,r)/(a*r)
```

```
        Y4 = (1.0D0 - 24.0D0/((a*r)**2))*cbessy(0,a,r)+
     1        (48.0D0/((a*r)**3) - 8.0D0/(a*r))*cbessy(1,a,r)
        d2cbessy = a*Y3/r + (2.0D0/r**2 - a**2)*Y2


      ELSEIF (n .eq. -1) THEN

        d2cbessy = a**2*cbessy(1,a,r) + a*cbessy(0,a,r)/r +
     1              2.0D0*cbessy(-1,a,r)/(r**2)

      ELSEIF (n .eq. -2) THEN

         Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
          d2cbessy = a**2*cbessy(0,a,r) + 3.0D0*a*cbessy(-1,a,r)/r +
     1                6.0D0*Y2/(r**2)


      ENDIF

      return
      end




      double complex function d1cbessi(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessi,d1cbessi

          d1cbessi = a*cbessi(n+1,a,r)+(n/r)*cbessi(n,a,r)

      return
      end
```

```
      double complex function d2cbessi(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex d2cbessi

        d2cbessi = (1.0,1.0)

      return
      end




      double complex function cbessh1(n,a,r)
      real*8 r
      complex*16 a
      integer n
          double complex cbessj,cbessy,cbessh1

          cbessh1 = cbessj(n,a,r)+(0.0, 1.0)*cbessy(n,a,r)

      return
      end




      double complex function cbessh2(n,a,r)
      real*8 r
      integer n
      complex*16 a
      double complex cbessj,cbessy,cbessh2

        cbessh2 = cbessj(n,a,r)-(0.0, 1.0)*cbessy(n,a,r)

      return
      end
```

```
      double complex function d1cbessh1(n,a,r)
      integer n
      real*8 r
      complex a
      double complex d1cbessj,d1cbessy,d1cbessh1

         d1cbessh1 = d1cbessj(n,a,r) + (0.0,1.0)*d1cbessy
     1             (n,a,r)

      return
      end




      double complex function d1cbessh2(n,a,r)
      integer n
      real*8 r
      complex a
      double complex d1cbessj,d1cbessy,d1cbessh2

         d1cbessh2 = d1cbessj(n,a,r) - (0.0,1.0)*d1cbessy
     1             (n,a,r)

      return
      end
```

LISTING FOR rf.f

```
C *********************************************************************
C
C               subprogram "rf.f"
C
C This subprogram was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C *********************************************************************


      SUBROUTINE ROD_POT(r,k,Om,n,cl_rod,ct_rod)


C EXTERNAL VARIABLES

      integer n
      real*8 r,k,Om
      complex*16 cl_rod,ct_rod

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      double complex cbessj
      double complex d1cbessj,d2cbessj

C INTERNAL VARIABLES

      complex*16 p,q


        p = zsqrt((Om**2/cl_rod**2) - k**2)
          d2_SP_rod = d2cbessj(n,p,r)
          d1_SP_rod = d1cbessj(n,p,r)
              SP_rod =   cbessj(n,p,r)
```

```
      q = zsqrt((Om**2/ct_rod**2) - k**2)
          d2_VXP_rod = d2cbessj(n,q,r)
          d1_VXP_rod = d1cbessj(n,q,r)
             VXP_rod =  cbessj(n,q,r)

          d2_VRTP_rod = d2cbessj(n+1,q,r)
          d1_VRTP_rod = d1cbessj(n+1,q,r)
             VRTP_rod =  cbessj(n+1,q,r)


      return
      end





      SUBROUTINE SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)

C EXTERNAL VARIABLES

      integer n
      real*8 k,ao_rod,ro,co,Om
      complex*16 sm(4,4)

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ******************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      common /OFLUID/ OFSC,d1_OFSC,M_OF

      complex*16 OFSC,d1_OFSC,M_OF

C ******************************************************************
```

```
C INTERNAL VARIABLES

        real*8 b2,k2
        integer n2

        b2 = ao_rod**2
        n2 =     n**2
        k2 =     k**2


        sm(1,1) = (lame_rod + 2.0D0*shear_rod)*d2_SP_rod +
     1     (lame_rod/ao_rod)*d1_SP_rod -
     1     (n2/b2 + k2)*lame_rod*SP_rod

        sm(1,2) = -2.0D0*n*shear_rod*VXP_rod/b2 +
     1      2.0D0*shear_rod*n*d1_VXP_rod/ao_rod

        sm(1,3) = (0.0D0,1.0D0)*2.0D0*shear_rod*k*d1_VRTP_rod

        sm(2,1) = (0.0D0,1.0D0)*2.0D0*shear_rod*k*d1_SP_rod

        sm(2,2) = (0.0D0,1.0D0)*k*n*VXP_rod*shear_rod/ao_rod

        sm(2,3) = shear_rod*(VRTP_rod*(n/b2 - k2 + 1.0D0/b2) -
     1     d1_VRTP_rod*(n/ao_rod + 1.0D0/ao_rod) - d2_VRTP_rod)

        sm(3,1) = 2.0D0*shear_rod*((n/b2)*SP_rod - (n/ao_rod)*
     1     d1_SP_rod)

        sm(3,2) = shear_rod*(-d2_VXP_rod + (1.0D0/ao_rod)*
     1     d1_VXP_rod - (n2/b2)*VXP_rod)

        sm(3,3) = (0.0D0,1.0D0)*(d1_VRTP_rod - VRTP_rod*
     1     (1.0D0/ao_rod + n/ao_rod))*k*shear_rod

C ***********************************************************************
C THIS TERM WILL CHANGE SIGN IF THE PRESSURE CONDITION IS
   REVERSED

        sm(1,4) =  +Om**2*ro*OFSC

C ***********************************************************************
        sm(2,4) = (0.0D0,0.0D0)

        sm(3,4) = (0.0D0,0.0D0)
```

```
      sm(4,1) = d1_SP_rod

      sm(4,2) = n*VXP_rod/ao_rod

      sm(4,3) = (0.0D0,1.0D0)*k*VRTP_rod

      sm(4,4) = -d1_OFSC


      return
      end




      SUBROUTINE ABC_ROD_SOLVE(exctype,sm)

C EXTERNAL VARIABLES

      integer exctype
      complex*16 sm(3,3)

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C INTERNAL VARIABLES

      complex*16 detA

C RADIAL EXCITATION exctype = 1
C AXIAL  EXCITATION exctype = 0

      if (exctype .eq. 1) then
        A1_rod =  sm(2,2)*sm(3,3) - sm(3,2)*sm(2,3)
        B1_rod = -sm(2,1)*sm(3,3) + sm(3,1)*sm(2,3)
        C1_rod =  sm(2,1)*sm(3,2) - sm(3,1)*sm(2,2)
      elseif (exctype .eq. 0) then
        A1_rod = -sm(1,2)*sm(3,3) - sm(3,2)*sm(1,3)
```

```
      B1_rod =  sm(1,1)*sm(3,3) + sm(3,1)*sm(1,3)
      C1_rod = -sm(1,1)*sm(3,2) - sm(3,1)*sm(1,2)
     endif

      detA = (sm(1,1)*sm(2,2)*sm(3,3) +
  1sm(1,2)*sm(2,3)*sm(3,1) + sm(2,1)*sm(3,2)*sm(1,3))-
  1(sm(3,1)*sm(2,2)*sm(1,3) + sm(1,1)*sm(3,2)*sm(2,3)+
  1sm(1,2)*sm(2,1)*sm(3,3))

      A1_rod = A1_rod/detA
      B1_rod = B1_rod/detA
      C1_rod = C1_rod/detA

     return
     end




     SUBROUTINE ABC_ROD_INVERT(exctype,sm)


C EXTERNAL VARIABLES

     integer exctype
     complex*16 sm(4,4)

     common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
  1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
  1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


     complex*16 SP_rod,d1_SP_rod,d2_SP_rod
     complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
     complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ******************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

     complex*16 OFSC,d1_OFSC,M_OF

     common /OFLUID/ OFSC,d1_OFSC,M_OF

C ******************************************************************
```

```
C INTERNAL VARIABLES

          integer n,iflag
          complex*16 sminv(4,4),work(4,8)

          n = 4
          iflag = 0

          CALL MINV(sm,sminv,work,n,iflag)



C RADIAL EXCITATION exctype = 1
C AXIAL  EXCITATION exctype = 0

          if (exctype .eq. 1) then
            A1_rod = -sminv(1,1)
            B1_rod = -sminv(2,1)
            C1_rod = -sminv(3,1)
             M_OF = -sminv(4,1)
          elseif (exctype .eq. 0) then
            A1_rod = -sminv(1,2)
            B1_rod = -sminv(2,2)
            C1_rod = -sminv(3,2)
             M_OF = -sminv(4,2)
          endif


          return
          end
```

```
      SUBROUTINE OUTPUT(tft,n,k,r,value)

C EXTERNAL VARIABLES

      integer n,tft
      real*8 k,r,zero
      complex*16 value

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C INTERNAL VARIABLES

      integer n2
      real*8 k2,r2,no,p11,p12
      complex*16 dill,dpp,exx,err,Srr,Sxx,ett,ur,vr,wr

      n2 = n**2
      r2 = r**2
      k2 = k**2

      zero = 1.D-20
      no  = 1.46D0
      p11  = .126D0
      p12  = .27D0


C  RADIAL STRESS/(Pr or Px)                                  tft = 0
C  LONGITUDINAL STRESS                                       tft = 1
C  AXIAL  DISPLACEMENT                                       tft = 2
C  THETA  DISPLACEMENT                                       tft = 3
C  RADIAL DISPLACEMENT                                       tft = 4
C  LONGITUDINAL STRAIN     e11/(Pr or Px)                    tft = 5
C  THETA  STRAIN           ett/(Pr or Px)                    tft = 6
C  RADIAL STRAIN           err/(Pr or Px)                    tft = 7
C                                                            tft = 8
C  OPTIC exx=0@k=0         ((dp/p)(r))/(Pr or Px)            tft = 9
C  OPTIC exx=const@k=0     ((dp/p)(r))/(Pr or Px)            tft = 10

```

```
        if (tft .eq. 0) then

        dill = A1_rod*(d2_SP_rod+d1_SP_rod/r -
     1       n2/r2*SP_rod - k2*SP_rod)
        Srr  = lame_rod*dill + 2.0D0*shear_rod*
     1      (A1_rod*d2_SP_rod - n/r2*B1_rod*VXP_rod +
     1      n/r*B1_rod*d1_VXP_rod + (0.0D0,1.0D0)*
     1      k*C1_rod*d1_VRTP_rod)

        value = Srr


C LONGITUDINAL STRESS USING (EQ 96)
        elseif(tft .eq. 1) then

           dill = A1_rod*(d2_SP_rod+d1_SP_rod/r -
     1          n2/r2*SP_rod - k2*SP_rod)

           Sxx = lame_rod*dill + 2.0D0*shear_rod*(
     1          -k2*A1_rod*SP_rod - (0.0D0,1.0D0)*k*C1_rod*
     1          (d1_VRTP_rod + VRTP_rod*(n + 1.0D0)/r))

           value = Sxx

        elseif(tft .eq. 2) then

           wr = (0.0D0,1.0D0)*k*A1_rod*SP_rod -
     1          C1_rod*(d1_VRTP_rod + 1.0D0/r*VRTP_rod + n/r*VRTP_rod)
           value = wr

        elseif(tft .eq. 3) then

           vr = -n*A1_rod*SP_rod/r - B1_rod*d1_VXP_rod +
     1          (0.0D0,1.0D0)*k*C1_rod*VRTP_rod
           value = vr

        elseif(tft .eq. 4) then

           ur = A1_rod*d1_SP_rod + n*B1_rod*VXP_rod/r +
     1        (0.0D0,1.0D0)*k*C1_rod*VRTP_rod
           value = ur

        elseif(tft .eq. 5) then

           exx = -k2*A1_rod*SP_rod -
     1          (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod -
```

```
     1            (0.0D0,1.0D0)*k*C1_rod*VRTP_rod*(1.0D0 + n)/r

        if (zabs(exx) .lt. dabs(zero)) then
          value = zero
        endif

        value = exx

      elseif(tft .eq. 6) then

        ett = A1_rod*(1.0D0/r*d1_SP_rod - n2/r2 * SP_rod) +
     1            B1_rod*(n/r2*VXP_rod - n/r * d1_VXP_rod) +
     1            (0.0D0,1.0D0)*C1_rod*k*VRTP_rod/r*(1.0D0 + n)

        if (zabs(ett) .lt. dabs(zero)) then
          value = zero
        endif
        value = ett

      elseif(tft .eq. 7) then

        err = A1_rod*d2_SP_rod +
     1            B1_rod*((-n/r2)*VXP_rod +
     1            (n/r)*d1_VXP_rod) +
     1            (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod

          if (zabs(err) .lt. dabs(zero)) then
            value = zero
          endif

        value = err

      elseif(tft .eq. 8) then
        value = 1.0D0
      elseif(tft .eq. 9) then

        exx = -k2*A1_rod*SP_rod -
     1            (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod -
     1            (0.0D0,1.0D0)*k*C1_rod*VRTP_rod*(1.0D0 + n)/r

          if (zabs(exx) .lt. dabs(zero)) then
            exx = zero
          endif

        err = A1_rod*d2_SP_rod +
     1            B1_rod*((-n/r2)*VXP_rod +
```

```
     1          (n/r)*d1_VXP_rod) +
     1          (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod

          if (zabs(err) .lt. dabs(zero)) then
            err = zero
          endif

        dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)

        value = dpp

      elseif(tft .eq. 10) then

        ett = A1_rod*(1.0D0/r*d1_SP_rod - n2/r2 * SP_rod) +
     1          B1_rod*(n/r2*VXP_rod - n/r * d1_VXP_rod) +
     1          (0.0D0,1.0D0)*C1_rod*k*VRTP_rod/r*(1.0D0 + n)

          if (zabs(ett) .lt. dabs(zero)) then
            value = zero
          endif

        err = A1_rod*d2_SP_rod +
     1          B1_rod*((-n/r2)*VXP_rod +
     1          (n/r)*d1_VXP_rod) +
     1          (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod

          if (zabs(err) .lt. dabs(zero)) then
            err = zero
          endif

        exx = -(lame_rod/(lame_rod + 2.0D0 * shear_rod))*(ett + err)
        dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
        value = dpp

      endif



      return
      end
```

```
C CMINV--Complex Matrix Inversion
C
        SUBROUTINE MINV (C,CINV,WORK,N,IFLAG)
C
C--Notes
C If IFLAG = 1, the matrix is singular
C Working precision (artificial zero) = 1D-12
C
C--External variables
        INTEGER N,IFLAG
        COMPLEX*16 C(N,N),CINV(N,N),WORK(N,2*N)
C
C--Internal variables
    INTEGER I,J,IP,IROW,JROW,JCOL,K
    COMPLEX*16 MAXPIV,S1,C1,SWITCH
    REAL*8 BMAG,T
C
    DO 110 I = 1, N, 1
      DO 100 J = 1, N, 1
        WORK(I,J) = C(I,J)
100     CONTINUE
110   CONTINUE
C
    DO 130 I = 1, N, 1
      DO 120 J = 1, N, 1
        WORK(I,J+N) = ( 0.0D0, 0.0D0 )
        IF (I.EQ.J) WORK(I,J+N) = ( 1.0D0, 0.0D0 )
120     CONTINUE
130   CONTINUE
C
    J = 1
    I = 1
140 IP = I
    MAXPIV = WORK(I,J)
C
    DO 150 IROW = I+1, N, 1
      S1 = WORK(IROW,J)
      IF (ZABS(S1) .LT. ZABS(MAXPIV)) GOTO 150
      IP = IROW
      MAXPIV = WORK(IROW,J)
150  CONTINUE
C
    IF (IP.EQ.1) GOTO 170
    DO 160 JROW = 1, 2*N, 1
      SWITCH = WORK(IP,JROW)
      WORK(IP,JROW) = WORK(I,JROW)
```

```
           WORK(I,JROW) = SWITCH
  160   CONTINUE
C
  170   BMAG = ZABS(MAXPIV)

C     WRITE(*,*)BMAG
C
        IF (BMAG .LT. 1.0D-12) GOTO 900
        DO 190 IROW = I+1, N, 1
          DO 180 JROW = J+1, 2*N, 1
            C1 = WORK(IROW,J)*WORK(I,JROW)/WORK(I,J)
            WORK(IROW,JROW) = WORK(IROW,JROW)-C1
  180     CONTINUE
          WORK(IROW,J) = ( 0., 0. )
  190   CONTINUE
        I = I+1
        J = J+1
C
        IF ((I.LT.N) .AND. (J.LT.N)) GOTO 140
C
        DO 210 I = 1, N, 1
          C1 = WORK(I,I)
          DO 200 J = I, 2*N, 1
            WORK(I,J) = WORK(I,J)/C1
  200     CONTINUE
  210   CONTINUE
C
        DO 240 I = N, 2, -1
          DO 230 J = I-1, 1, -1
            C1 = WORK(J,I)
            DO 220 JCOL = J+1, 2*N, 1
              WORK(J,JCOL) = WORK(J,JCOL)-C1*WORK(I,JCOL)
  220       CONTINUE
  230     CONTINUE
  240   CONTINUE
C
        DO 260 I = 1, N, 1
          DO 250 J = 1, N, 1
            CINV(I,J) = WORK(I,J+N)
  250     CONTINUE
  260   CONTINUE
        IFLAG = 0
C
        T = 0.0D0
        K = 0
CC--The norm of the matrix inversion is T
```

```
CC     DO 190 I = 1, N, 1
CC       DO 190 J = 1, N, 1
CC         C1 = ( 0., 0. )
CC         DO 180 K = 1, N, 1
CC           C1 = C1+C(I,K)*WORK(K,J)
CC180        CONTINUE
CC         S1 = ( -1., 0. )*C1
CC         IF (I.EQ.J) S1 = S1 + ( 1., 0. )
CC         T = T + (REAL(S1)**2) + (IMAG(S1)**2)
CC190  CONTINUE
CC     T = SQRT(T)
C
    RETURN
C
900  IFLAG = 1


    WRITE(*,*)'BMAG < ARTIFICIAL ZERO (1.0D-12) RETURNING FROM MINV'
    RETURN
    END
```

LISTING FOR c1.f

```
C ********************************************************************
C
C               subprogram "c1.f"
C
C This subprogram was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C ********************************************************************


        SUBROUTINE C1A_POT(r,k,Om,n)


C EXTERNAL VARIABLES

        integer n
        real*8 r,k,Om


C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

        complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
        complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
        complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
        complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

        complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
        complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
        complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
        complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

        complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
        complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
        complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
        complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

        complex*16 lame_c1,shear_c1,cl_c1,ct_c1

        complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

        common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1          SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1          SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
```

```
     1           SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1           VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1           VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1           VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1           VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1           VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1           VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1           VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1           VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1           lame_c1,shear_c1,cl_c1,ct_c1,
     1           A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C ****************************************************************

      double complex cbessj,cbessy
      double complex d1cbessj,d2cbessj
      double complex d1cbessy,d2cbessy

C INTERNAL VARIABLES

      complex*16 p,q

C DISPLACEMENT  SCALAR POTENTIALS ARE EVALUATED

        p = zsqrt((Om**2/cl_c1**2) - k**2)
          d2_SP_CY1_a1 = d2cbessj(n,p,r)
          d1_SP_CY1_a1 = d1cbessj(n,p,r)
             SP_CY1_a1 =   cbessj(n,p,r)
          d2_SP_CY1_a2 = d2cbessy(n,p,r)
          d1_SP_CY1_a2 = d1cbessy(n,p,r)
             SP_CY1_a2 =   cbessy(n,p,r)

        q = zsqrt((Om**2/ct_c1**2) - k**2)
          d2_VXP_CY1_a1 = d2cbessj(n,q,r)
          d1_VXP_CY1_a1 = d1cbessj(n,q,r)
             VXP_CY1_a1 =   cbessj(n,q,r)
          d2_VXP_CY1_a2 = d2cbessy(n,q,r)
          d1_VXP_CY1_a2 = d1cbessy(n,q,r)
             VXP_CY1_a2 =   cbessy(n,q,r)

        d2_VRTP_CY1_a1 = d2cbessj(n+1,q,r)
        d1_VRTP_CY1_a1 = d1cbessj(n+1,q,r)
           VRTP_CY1_a1 =   cbessj(n+1,q,r)
        d2_VRTP_CY1_a2 = d2cbessy(n+1,q,r)
        d1_VRTP_CY1_a2 = d1cbessy(n+1,q,r)
           VRTP_CY1_a2 =   cbessy(n+1,q,r)
```

```
      return
      end



      SUBROUTINE C1B_POT(r,k,Om,n)


C EXTERNAL VARIABLES

      integer n
      real*8 r,k,Om


C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1

      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1       SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1       SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1       SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1       VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1       VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1       VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1       VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1       VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
```

```
     1           VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1           VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1           VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1           lame_c1,shear_c1,cl_c1,ct_c1,
     1           A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C ******************************************************************

      double complex cbessj,cbessy
      double complex d1cbessj,d2cbessj
      double complex d1cbessy,d2cbessy

C INTERNAL VARIABLES

      complex*16 p,q

C DISPLACEMENT  SCALAR POTENTIALS ARE EVALUATED

        p = zsqrt((Om**2/cl_c1**2) - k**2)
           d2_SP_CY1_b1 = d2cbessj(n,p,r)
           d1_SP_CY1_b1 = d1cbessj(n,p,r)
              SP_CY1_b1 =   cbessj(n,p,r)
           d2_SP_CY1_b2 = d2cbessy(n,p,r)
           d1_SP_CY1_b2 = d1cbessy(n,p,r)
              SP_CY1_b2 =   cbessy(n,p,r)


        q = zsqrt((Om**2/ct_c1**2) - k**2)
           d2_VXP_CY1_b1 = d2cbessj(n,q,r)
           d1_VXP_CY1_b1 = d1cbessj(n,q,r)
              VXP_CY1_b1 =   cbessj(n,q,r)
           d2_VXP_CY1_b2 = d2cbessy(n,q,r)
           d1_VXP_CY1_b2 = d1cbessy(n,q,r)
              VXP_CY1_b2 =   cbessy(n,q,r)

         d2_VRTP_CY1_b1 = d2cbessj(n+1,q,r)
         d1_VRTP_CY1_b1 = d1cbessj(n+1,q,r)
            VRTP_CY1_b1 =   cbessj(n+1,q,r)
         d2_VRTP_CY1_b2 = d2cbessy(n+1,q,r)
         d1_VRTP_CY1_b2 = d1cbessy(n+1,q,r)
            VRTP_CY1_b2 =   cbessy(n+1,q,r)



      return
      end
```

```
      SUBROUTINE SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl,bo_c1,
     1Om,ro,co,smrc1)

C EXTERNAL VARIABLES

      integer n
      real*8 k,ao_rod,ao_1cyl,bo_c1,Om,ro,co
      complex*16 smrc1(10,10)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1

      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1          SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1          SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1          SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1          VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1          VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1          VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1          VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1          VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1          VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
```

```
     1              VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1              VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1              lame_c1,shear_c1,cl_c1,ct_c1,
     1              A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C *****************************************************************

C DEFINITION FOR COMMON BLOCK /ROD/

        common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


        complex*16 SP_rod,d1_SP_rod,d2_SP_rod
        complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
        complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
        complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        common /OFLUID/ OFSC,d1_OFSC,M_OF

        complex*16 OFSC,d1_OFSC,M_OF

C *****************************************************************

C INTERNAL VARIABLES

        complex*16 L2GC1,L2GR
        real*8 b2,k2,a2
        integer n2

        a2    = ao_rod**2
        b2    = bo_c1**2
        n2    = n**2
        k2    = k**2
        L2GC1  = lame_c1 + 2.0D0 * shear_c1
        L2GR   = lame_rod+ 2.0D0 * shear_rod

C BOUNDARY CONDITION #1 (EQ 113)

        smrc1(1,1) = L2GC1*d2_SP_CY1_b1 + (lame_c1/bo_c1)*
     1d1_SP_CY1_b1 - lame_c1*SP_CY1_b1*(n2/b2 + k2)
```

```
      smrc1(1,2) = L2GC1*d2_SP_CY1_b2 + (lame_c1/bo_c1)*
     1d1_SP_CY1_b2 - lame_c1*SP_CY1_b2*(n2/b2 + k2)

      smrc1(1,3) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b1-bo_c1*
     1d1_VXP_CY1_b1)

      smrc1(1,4) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b2-bo_c1*
     1d1_VXP_CY1_b2)

      smrc1(1,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b1

      smrc1(1,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b2

      smrc1(1,7) = (0.0D0,0.0D0)

      smrc1(1,8) = (0.0D0,0.0D0)

      smrc1(1,9) = (0.0D0,0.0D0)

      smrc1(1,10)= +ro*Om**2*OFSC

C BOUNDARY CONDITION #2 (EQ 116)

      smrc1(2,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b1

      smrc1(2,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b2

      smrc1(2,3) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b1/bo_c1

      smrc1(2,4) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b2/bo_c1

      smrc1(2,5) = shear_c1*(VRTP_CY1_b1*(n/b2 - k2 +
     1 1.0D0/b2) - d1_VRTP_CY1_b1*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b1)

      smrc1(2,6) = shear_c1*(VRTP_CY1_b2*(n/b2 - k2 +
     1 1.0D0/b2) - d1_VRTP_CY1_b2*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b2)

      smrc1(2,7) = (0.0D0,0.0D0)

      smrc1(2,8) = (0.0D0,0.0D0)

      smrc1(2,9) = (0.0D0,0.0D0)

      smrc1(2,10) = (0.0D0,0.0D0)
```

```
C BOUNDARY CONDITION #3 (EQ 120)

      smrc1(3,1) = (shear_c1*2.0D0*n/bo_c1)*((1.0D0/bo_c1)*
     1SP_CY1_b1 - d1_SP_CY1_b1)

      smrc1(3,2) = (shear_c1*2.0D0*n/bo_c1)*((1.0D0/bo_c1)*
     1SP_CY1_b2 - d1_SP_CY1_b2)

      smrc1(3,3) = shear_c1*(-d2_VXP_CY1_b1 + 1.0D0/bo_c1*
     1d1_VXP_CY1_b1 - n2/b2*VXP_CY1_b1)

      smrc1(3,4) = shear_c1*(-d2_VXP_CY1_b2 + 1.0D0/bo_c1*
     1d1_VXP_CY1_b2 - n2/b2*VXP_CY1_b2)

      smrc1(3,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_b1
     1- VRTP_CY1_b1*(1.0D0 + n)/bo_c1)

      smrc1(3,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_b2
     1- VRTP_CY1_b2*(1.0D0 + n)/bo_c1)

      smrc1(3,7) = (0.0D0,0.0D0)

      smrc1(3,8) = (0.0D0,0.0D0)

      smrc1(3,9) = (0.0D0,0.0D0)

      smrc1(3,10) = (0.0D0,0.0D0)


C BOUNDARY CONDITION #4 (EQ 123)

      smrc1(4,1) = d1_SP_CY1_b1

      smrc1(4,2) = d1_SP_CY1_b2

      smrc1(4,3) = n/bo_c1*VXP_CY1_b1

      smrc1(4,4) = n/bo_c1*VXP_CY1_b2

      smrc1(4,5) = (0.0D0,1.0D0)*k*VRTP_CY1_b1

      smrc1(4,6) = (0.0D0,1.0D0)*k*VRTP_CY1_b2

      smrc1(4,7) = (0.0D0,0.0D0)

      smrc1(4,8) = (0.0D0,0.0D0)
```

```
      smrc1(4,9) = (0.0D0,0.0D0)

      smrc1(4,10)= -d1_OFSC

C BOUNDARY CONDITION #5 (EQ 125)

      smrc1(5,1) = L2GC1*d2_SP_CY1_a1 + lame_c1/ao_1cyl
     1*d1_SP_CY1_a1 - lame_c1*SP_CY1_a1*(n2/a2 + k2)

      smrc1(5,2) = L2GC1*d2_SP_CY1_a2 + lame_c1/ao_1cyl
     1*d1_SP_CY1_a2 - lame_c1*SP_CY1_a2*(n2/a2 + k2)

      smrc1(5,3) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a1 - ao_1cyl
     1*d1_VXP_CY1_a1)

      smrc1(5,4) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a2 - ao_1cyl
     1*d1_VXP_CY1_a2)

      smrc1(5,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a1

      smrc1(5,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a2

      smrc1(5,7) = -L2GR*d2_SP_rod - lame_rod/ao_rod*
     1d1_SP_rod + lame_rod*SP_rod*(n2/a2 + k2)

      smrc1(5,8) = -2.0D0*shear_rod*n/ao_rod*(d1_VXP_rod - 1.0D0/
     1ao_rod*VXP_rod)

      smrc1(5,9) = (0.0D0,-1.0D0)*k*2.0D0*shear_rod*d1_VRTP_rod

      smrc1(5,10)= (0.0D0,0.0D0)

C BOUNDARY CONDITION #6 (EQ 127)

      smrc1(6,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a1

      smrc1(6,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a2

      smrc1(6,3) = (0.0D0,1.0D0)*k*n*shear_c1/ao_rod*VXP_CY1_a1

      smrc1(6,4) = (0.0D0,1.0D0)*k*n*shear_c1/ao_rod*VXP_CY1_a2

      smrc1(6,5) = shear_c1*(VRTP_CY1_a1*((n+1.0D0)/a2 - k2)
     1- d1_VRTP_CY1_a1*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a1)
```

```
      smrc1(6,6) = shear_c1*(VRTP_CY1_a2*((n+1.0D0)/a2 - k2)
     1- d1_VRTP_CY1_a2*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a2)

      smrc1(6,7) = (0.0D0,-1.0D0)*k*2.0*shear_rod*d1_SP_rod

      smrc1(6,8) = (0.0D0,-1.0D0)*k*n*shear_rod/ao_rod*VXP_rod

      smrc1(6,9) = -shear_rod*(VRTP_rod*((n+1.0D0)/a2 - k2)
     1- d1_VRTP_rod*(n+1.0D0)/ao_rod - d2_VRTP_rod)

      smrc1(6,10) = (0.0D0,0.0D0)

C BOUNDARY CONDITION #7 (EQ 129)

      smrc1(7,1) = shear_c1*2.0D0*n/ao_1cyl*(SP_CY1_a1/ao_1cyl
     1- d1_SP_CY1_a1)

      smrc1(7,2) = shear_c1*2.0D0*n/ao_1cyl*(SP_CY1_a2/ao_1cyl
     1- d1_SP_CY1_a2)

      smrc1(7,3) = shear_c1*(-d2_VXP_CY1_a1 + d1_VXP_CY1_a1/
     1 ao_1cyl - n2/a2*VXP_CY1_a1)

      smrc1(7,4) = shear_c1*(-d2_VXP_CY1_a2 + d1_VXP_CY1_a2/
     1 ao_1cyl - n2/a2*VXP_CY1_a2)

      smrc1(7,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a1 -
     1 VRTP_CY1_a1*(1.0D0+n)/ao_1cyl)

      smrc1(7,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a2 -
     1 VRTP_CY1_a2*(1.0D0+n)/ao_1cyl)

      smrc1(7,7) = -2.0D0*n*shear_rod/a2*(SP_rod - ao_rod*d1_SP_rod)

      smrc1(7,8) = -shear_rod*(-d2_VXP_rod + d1_VXP_rod/ao_rod
     1 - n2/a2*VXP_rod)

      smrc1(7,9) = (0.0D0,-1.0D0)*k*shear_rod*(d1_VRTP_rod
     1 - VRTP_rod*(1.0D0 + n)/ao_rod)

      smrc1(7,10)= (0.0D0,0.0D0)

C BOUNDARY CONDITION #8 (EQ 132)

      smrc1(8,1) = d1_SP_cy1_a1
```

```
          smrc1(8,2) = d1_SP_cy1_a2
          smrc1(8,3) = n*VXP_CY1_a1/ao_1cyl
          smrc1(8,4) = n*VXP_CY1_a2/ao_1cyl
          smrc1(8,5) = (0.0D0,1.0D0)*k*VRTP_CY1_a1
          smrc1(8,6) = (0.0D0,1.0D0)*k*VRTP_CY1_a2
          smrc1(8,7) = -d1_SP_rod
          smrc1(8,8) = -n*VXP_rod/ao_rod
          smrc1(8,9) = (0.0D0,-1.0D0)*k*VRTP_rod
          smrc1(8,10) = (0.0D0,0.0D0)
C BOUNDARY CONDITION #9 (EQ 135)
          smrc1(9,1) = -n*SP_CY1_a1/ao_1cyl
          smrc1(9,2) = -n*SP_CY1_a2/ao_1cyl
          smrc1(9,3) = -d1_VXP_CY1_a1
          smrc1(9,4) = -d1_VXP_CY1_a2
          smrc1(9,5) = (0.0D0,1.0D0)*k*VRTP_CY1_a1
          smrc1(9,6) = (0.0D0,1.0D0)*k*VRTP_CY1_a2
          smrc1(9,7) = n*SP_rod/ao_rod
          smrc1(9,8) = d1_VXP_rod
          smrc1(9,9)= (0.0D0,-1.0D0)*k*VRTP_rod
          smrc1(9,10)= (0.0D0,0.0D0)
C BOUNDARY CONDITION #10 (EQ 138)
          smrc1(10,1) = (0.0D0,1.0D0)*k*SP_CY1_a1
          smrc1(10,2) = (0.0D0,1.0D0)*k*SP_CY1_a2
```

```
      smrc1(10,3) = (0.0D0,0.0D0)

      smrc1(10,4) = (0.0D0,0.0D0)

      smrc1(10,5) = -d1_VRTP_CY1_a1 - (n + 1.0D0)*VRTP_CY1_a1
     1/ao_1cyl

      smrc1(10,6) = -d1_VRTP_CY1_a2 - (n + 1.0D0)*VRTP_CY1_a2
     1/ao_1cyl

      smrc1(10,7) = (0.0D0,-1.0D0)*k*SP_rod

      smrc1(10,8) = (0.0D0,0.0D0)

      smrc1(10,9) = d1_VRTP_rod + VRTP_rod*(n + 1.0D0)
     1/ao_rod

      smrc1(10,10) = (0.0D0,0.0D0)


      return
      end




      SUBROUTINE ABC_RC1_INVERT(n,exctype,smrc1,b)


C EXTERNAL VARIABLES

      integer n,exctype
      real*8 b
      complex*16 smrc1(10,10)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
```

```
      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1          SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1          SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1          SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1          VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1          VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1          VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1          VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1          VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1          VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1          VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1          VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1          lame_c1,shear_c1,cl_c1,ct_c1,
     1          A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /ROD/

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      complex*16 OFSC,d1_OFSC,M_OF

      common /OFLUID/ OFSC,d1_OFSC,M_OF

```

```
C ******************************************************************

C INTERNAL VARIABLES

        integer size,iflag
        complex*16 smrc1inv(10,10),workrc1(10,20)

        size = 10
        iflag = 0


        CALL MINV(smrc1,smrc1inv,workrc1,size,iflag)



C RADIAL EXCITATION exctype = 1
C AXIAL  EXCITATION exctype = 0

        if (exctype .eq. 1) then
            A1_C1 = -smrc1inv(1,1)
            A2_C1 = -smrc1inv(2,1)
            B1_C1 = -smrc1inv(3,1)
            B2_C1 = -smrc1inv(4,1)
            C1_C1 = -smrc1inv(5,1)
            C2_C1 = -smrc1inv(6,1)
            A1_rod = -smrc1inv(7,1)
            B1_rod = -smrc1inv(8,1)
            C1_rod = -smrc1inv(9,1)
            M_OF = -smrc1inv(10,1)
        elseif (exctype .eq. 0) then
            A1_C1 = -smrc1inv(1,2)
            A2_C1 = -smrc1inv(2,2)
            B1_C1 = -smrc1inv(3,2)
            B2_C1 = -smrc1inv(4,2)
            C1_C1 = -smrc1inv(5,2)
            C2_C1 = -smrc1inv(6,2)
            A1_rod = -smrc1inv(7,2)
            B1_rod = -smrc1inv(8,2)
            C1_rod = -smrc1inv(9,2)
            M_OF = -smrc1inv(10,2)
        endif


          return
          end
```

```
      SUBROUTINE OUTPUT_RC1(tft,n,k,r,value)

C EXTERNAL VARIABLES

      integer n,tft
      real*8 k,r,zero
      complex*16 value

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1            SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1            SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1            SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1            VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1            VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1            VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1            VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1            VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1            VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1            VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1            VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1            lame_c1,shear_c1,cl_c1,ct_c1,
     1            A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
```

```

C ******************************************************************

C DEFINITIONS FOR COMMON BLOCK /ROD/

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ******************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      common /OFLUID/ OFSC,d1_OFSC,M_OF

      complex*16 OFSC,d1_OFSC,M_OF

C ******************************************************************

C INTERNAL VARIABLES

      integer n2
      real*8 k2,r2,no,p11,p12
      complex*16 dpp,exx,err,Srr,Sxx,ett,uc,vc,wc
      complex*16 A1C1,A2C1,B1C1,B2C1,C1C1,C2C1,L2GC1

      n2 = n**2
      r2 = r**2
      k2 = k**2

      zero  = 1.000D-30
      no    = 1.460D0
      p11   = 0.126D0
      p12   = 0.270D0
      L2GC1 = lame_c1 + 2.0D0*shear_c1


```

```
C RADIAL STRESS/(Pr or Px)                                    tft = 0
C                                                             tft = 1
C AXIAL  DISPLACEMENT                                          tft = 2
C THETA  DISPLACEMENT                                          tft = 3
C RADIAL DISPLACEMENT                                          tft = 4
C LONGITUDINAL STRAIN     e11/(Pr or Px)                       tft = 5
C THETA  STRAIN           ett/(Pr or Px)                       tft = 6
C RADIAL STRAIN           err/(Pr or Px)                       tft = 7
C                                                              tft = 8
C OPTIC exx=0@k=0         ((dp/p)(r))/(Pr or Px)               tft = 9
C OPTIC exx=const@k=0     ((dp/p)(r))/(Pr or Px)               tft = 10




        if (tft .eq. 0) then

          A1C1 = A1_C1*(L2GC1*d2_SP_CY1_a1 + lame_c1*
     1           d1_SP_CY1_a1/r - lame_c1*SP_CY1_a1*(n2/
     1           r2 + k2))

          A2C1 = A2_C1*(L2GC1*d2_SP_CY1_a2 + lame_c1*
     1           d1_SP_CY1_a2/r - lame_c1* SP_CY1_a2*(n2/
     1           r2 + k2))

          B1C1 = B1_C1*-2.0*shear_c1*n/r**2*(VXP_CY1_a1 -
     1           r*d1_VXP_CY1_a1)

          B2C1 = B2_C1*-2.0*shear_c1*n/r**2*(VXP_CY1_a2 -
     1           r*d1_VXP_CY1_a2)

          C1C1 = (0.0D0,1.0D0)*C1_C1*2.0D0*shear_c1*k*d1_VRTP_CY1_a1

          C2C1 = (0.0D0,1.0D0)*C2_C1*2.0D0*shear_c1*k*d1_VRTP_CY1_a2

          Srr = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

          value = Srr


C LONGITUDINAL STRESS USING (EQ 153)
        elseif(tft .eq. 1) then

          A1C1 = A1_C1*(lame_c1*(d2_SP_CY1_a1 + d1_SP_CY1_a1/r)
     1         - SP_CY1_a1*(lame_c1*n2/r2 + k2*(lame_c1
     1         + 2.0D0*shear_c1)))
```

```
      A2C1 = A2_C1*(lame_c1*(d2_SP_CY1_a2 + d1_SP_CY1_a2/r)
     1     - SP_CY1_a2*(lame_c1*n2/r2 + k2*(lame_c1
     1     + 2.0D0*shear_c1)))

      C1C1 = C1_C1*((0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a1 +
     1          VRTP_CY1_a1*(n + 1.0D0)/r))*2.0D0*shear_c1

      C2C1 = C2_C1*((0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a2 +
     1          VRTP_CY1_a2*(n + 1.0D0)/r))*2.0D0*shear_c1

      Sxx = A1C1 + A2C1 + C1C1 + C2C1

      value = Sxx


C AXIAL DISPLACEMENT USING (EQ 149)
      elseif(tft .eq. 2) then

      A1C1 = A1_C1*(0.0D0,1.0D0)*k*SP_CY1_a1

      A2C1 = A2_C1*(0.0D0,1.0D0)*k*SP_CY1_a2

      C1C1 = C1_C1*(-d1_VRTP_CY1_a1 - VRTP_CY1_a1*
     1          (n + 1.0D0)/r)

      C2C1 = C2_C1*(-d1_VRTP_CY1_a2 - VRTP_CY1_a2*
     1          (n + 1.0D0)/r)

      wc = A1C1 + A2C1 + C1C1 + C2C1

      value = wc

C THETA DISPLACEMENT USING (EQ 150)
      elseif(tft .eq. 3) then


      A1C1 = A1_C1*-SP_CY1_a1*n/r

      A2C1 = A2_C1*-SP_CY1_a2*n/r

      B1C1 = -B1_C1*d1_VXP_CY1_a1

      B2C1 = -B2_C1*d1_VXP_CY1_a2

      C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1
```

```
            C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2

            vc  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            value = vc

C RADIAL DISPLACEMENT USING (EQ 148)
        elseif(tft .eq. 4) then

            A1C1 = A1_C1*d1_SP_CY1_a1

            A2C1 = A2_C1*d1_SP_CY1_a2

            B1C1 = B1_C1*VXP_CY1_a1*n/r

            B2C1 = B2_C1*VXP_CY1_a2*n/r

            C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1

            C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2

            uc  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            value = uc

C LONGITUDINAL STRAIN USING (EQ 147)
        elseif(tft .eq. 5) then

            A1C1 = -A1_C1*k2*SP_CY1_a1

            A2C1 = -A2_C1*k2_SP_CY1_a2

            C1C1 = C1_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a1 +
     1             VRTP_CY1_a1*(n + 1.0D0)/r)

            C2C1 = C2_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a2 +
     1             VRTP_CY1_a2*(n + 1.0D0)/r)

            exx  = A1C1 + A2C1 + C1C1 + C2C1


              if (zabs(exx) .lt. dabs(zero)) then
                value = zero
              endif

```

```
          value = exx

C THETA STRAIN USING (EQ 145)
        elseif(tft .eq. 6) then

          A1C1 = A1_C1*(d1_SP_CY1_a1/r - SP_CY1_a1*n2/r2)

          A2C1 = A2_C1*(d1_SP_CY1_a2/r - SP_CY1_a2*n2/r2)

          B1C1 = B1_C1*(-d1_VXP_CY1_a1*n/r + VXP_CY1_a1*n/r2)

          B2C1 = B2_C1*(-d1_VXP_CY1_a2*n/r + VXP_CY1_a2*n/r2)

          C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1*(n + 1.0D0)/r

          C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2*(n + 1.0D0)/r

          ett  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            if (zabs(ett) .lt. dabs(zero)) then
              value = zero
            endif

          value = ett

C RADIAL STRAIN USING (eq 143)
        elseif(tft .eq. 7) then

          A1C1 = A1_C1*d2_SP_CY1_a1

          A2C1 = A2_C1*d2_SP_CY1_a2

          B1C1 = B1_C1*(d1_VXP_CY1_a1 - VXP_CY1_a1/r)*n/r

          B2C1 = B2_C1*(d1_VXP_CY1_a2 - VXP_CY1_a2/r)*n/r

          C1C1 = C1_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a1

          C2C1 = C2_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a2

          err  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            if (zabs(err) .lt. dabs(zero)) then
              value = zero
            endif
```

```
          value = err

C UNUSED AT PRESENT
        elseif(tft .eq. 8) then
          value = 1.0

C OPTIC dp/p/uPa USING (EQ 147 & 143)
C exx = 0 @ k = 0
        elseif(tft .eq. 9) then

          A1C1 = -A1_C1*k2*SP_CY1_a1

          A2C1 = -A2_C1*k2_SP_CY1_a2

          C1C1 = C1_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a1 +
     1           VRTP_CY1_a1*(n + 1.0D0)/r)

          C2C1 = C2_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a2 +
     1           VRTP_CY1_a2*(n + 1.0D0)/r)

          exx  = A1C1 + A2C1 + C1C1 + C2C1

            if (zabs(exx) .lt. dabs(zero)) then
              exx = zero
            endif

          A1C1 = A1_C1*d2_SP_CY1_a1

          A2C1 = A2_C1*d2_SP_CY1_a2

          B1C1 = B1_C1*(d1_VXP_CY1_a1 - VXP_CY1_a1/r)*n/r

          B2C1 = B2_C1*(d1_VXP_CY1_a2 - VXP_CY1_a2/r)*n/r

          C1C1 = C1_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a1

          C2C1 = C2_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a2

          err  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            if (zabs(err) .lt. dabs(zero)) then
              err = zero
            endif

          dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
```

```
          value = dpp

C OPTIC dp/p/uPa USING (EQ 145 & 143)
C exx = constant @ k = 0
        elseif(tft .eq. 10) then

          A1C1 = A1_C1*(d1_SP_CY1_a1/r - SP_CY1_a1*n2/r2)

          A2C1 = A2_C1*(d1_SP_CY1_a2/r - SP_CY1_a2*n2/r2)

          B1C1 = B1_C1*(-d1_VXP_CY1_a1*n/r + VXP_CY1_a1*n/r2)

          B2C1 = B2_C1*(-d1_VXP_CY1_a2*n/r + VXP_CY1_a2*n/r2)

          C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1*(n + 1.0D0)/r

          C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2*(n + 1.0D0)/r

          ett  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            if (zabs(ett) .lt. dabs(zero)) then
              value = zero
            endif

          A1C1 = A1_C1*d2_SP_CY1_a1

          A2C1 = A2_C1*d2_SP_CY1_a2

          B1C1 = B1_C1*(d1_VXP_CY1_a1 - VXP_CY1_a1/r)*n/r

          B2C1 = B2_C1*(d1_VXP_CY1_a2 - VXP_CY1_a2/r)*n/r

          C1C1 = C1_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a1

          C2C1 = C2_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a2

          err  = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1

            if (zabs(err) .lt. dabs(zero)) then
              err = zero
            endif

          exx = -(lame_rod/(lame_rod + 2.0D0 * shear_rod))*(ett + err)
          dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
          value = dpp

```

```
endif


return
end
```

LISTING FOR c2.f

```
C ******************************************************************
C
C              subprogram "c2.f"
C
C This subprogram was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C ******************************************************************
C Modifications on 5-14-95; Removed K & I Bessel functions.
C The routine is now based on J & Y only, using series to handle
C complex arguments. C2B_POT, C2C_POT


      SUBROUTINE C2B_POT(r,k,Om,n)


C EXTERNAL VARIABLES

      integer n
      real*8 r,k,Om


C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

```

```
      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
     1         SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
     1         SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
     1         SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
     1         VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
     1         VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
     1         VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
     1         VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
     1         VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
     1         VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
     1         VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
     1         VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
     1         lame_c2,shear_c2,cl_c2,ct_c2,
     1         A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C ******************************************************************

      double complex cbessj,cbessy
      double complex d1cbessj,d2cbessj
      double complex d1cbessy,d2cbessy


C INTERNAL VARIABLES

      complex*16 p,q

C DISPLACEMENT  SCALAR POTENTIALS ARE EVALUATED

      p = zsqrt((Om**2/cl_c2**2) - k**2)
          d2_SP_CY2_b1 = d2cbessj(n,p,r)
          d1_SP_CY2_b1 = d1cbessj(n,p,r)
             SP_CY2_b1 =   cbessj(n,p,r)
          d2_SP_CY2_b2 = d2cbessy(n,p,r)
          d1_SP_CY2_b2 = d1cbessy(n,p,r)
             SP_CY2_b2 =   cbessy(n,p,r)

      q = zsqrt((Om**2/ct_c2**2) - k**2)
          d2_VXP_CY2_b1 = d2cbessj(n,q,r)
          d1_VXP_CY2_b1 = d1cbessj(n,q,r)
             VXP_CY2_b1 =   cbessj(n,q,r)
          d2_VXP_CY2_b2 = d2cbessy(n,q,r)
          d1_VXP_CY2_b2 = d1cbessy(n,q,r)
             VXP_CY2_b2 =   cbessy(n,q,r)
```

```
      d2_VRTP_CY2_b1 = d2cbessj(n+1,q,r)
      d1_VRTP_CY2_b1 = d1cbessj(n+1,q,r)
         VRTP_CY2_b1 =   cbessj(n+1,q,r)
      d2_VRTP_CY2_b2 = d2cbessy(n+1,q,r)
      d1_VRTP_CY2_b2 = d1cbessy(n+1,q,r)
         VRTP_CY2_b2 =   cbessy(n+1,q,r)





      return
      end



      SUBROUTINE C2C_POT(r,k,Om,n)


C EXTERNAL VARIABLES

      integer n
      real*8 r,k,Om


C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/


      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
```

```


      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
     1          SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
     1          SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
     1          SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
     1          VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
     1          VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
     1          VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
     1          VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
     1          VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
     1          VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
     1          VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
     1          VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
     1          lame_c2,shear_c2,cl_c2,ct_c2,
     1          A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C ******************************************************************


      double complex cbessj,cbessy
      double complex d1cbessj,d2cbessj
      double complex d1cbessy,d2cbessy

C INTERNAL VARIABLES

      complex*16 p,q

C DISPLACEMENT  SCALAR POTENTIALS ARE EVALUATED

      p = zsqrt((Om**2/cl_c2**2) - k**2)
         d2_SP_CY2_c1 = d2cbessj(n,p,r)
         d1_SP_CY2_c1 = d1cbessj(n,p,r)
            SP_CY2_c1 =   cbessj(n,p,r)
         d2_SP_CY2_c2 = d2cbessy(n,p,r)
         d1_SP_CY2_c2 = d1cbessy(n,p,r)
            SP_CY2_c2 =   cbessy(n,p,r)


      q = zsqrt((Om**2/ct_c2**2) - k**2)
         d2_VXP_CY2_c1 = d2cbessj(n,q,r)
         d1_VXP_CY2_c1 = d1cbessj(n,q,r)
            VXP_CY2_c1 =   cbessj(n,q,r)
         d2_VXP_CY2_c2 = d2cbessy(n,q,r)
         d1_VXP_CY2_c2 = d1cbessy(n,q,r)
            VXP_CY2_c2 =   cbessy(n,q,r)
```

```
          d2_VRTP_CY2_c1 = d2cbessj(n+1,q,r)
          d1_VRTP_CY2_c1 = d1cbessj(n+1,q,r)
             VRTP_CY2_c1 =  cbessj(n+1,q,r)
          d2_VRTP_CY2_c2 = d2cbessy(n+1,q,r)
          d1_VRTP_CY2_c2 = d1cbessy(n+1,q,r)
             VRTP_CY2_c2 =  cbessy(n+1,q,r)



      return
      end



      SUBROUTINE SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2,
     1Om,ro,co,ri,ci,smrc2)

C EXTERNAL VARIABLES

      integer n
      real*8 k,ao_1cyl,bo_c1,co_c2,Om,ro,co,ri,ci
      complex*16 smrc2(13,13)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1


      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
```

```
     1              SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1              SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1              SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1              VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1              VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1              VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1              VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1              VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1              VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1              VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1              VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1              lame_c1,shear_c1,cl_c1,ct_c1,
     1              A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C ****************************************************************



C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/


      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2


      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
     1              SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
     1              SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
     1              SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
     1              VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
```

```
     1            VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
     1            VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
     1            VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
     1            VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
     1            VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
     1            VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
     1            VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
     1            lame_c2,shear_c2,cl_c2,ct_c2,
     1            A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C ******************************************************************



C DEFINITION FOR COMMON BLOCK /ROD/

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ******************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      common /OFLUID/ OFSC,d1_OFSC,M_OF

      complex*16 OFSC,d1_OFSC,M_OF

C ******************************************************************

C ******************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

      common /IFLUID/ IFSC,d1_IFSC,D_IF

      complex*16 IFSC,d1_IFSC,D_IF

C ******************************************************************
```

```
C INTERNAL VARIABLES

      complex*16 L2GC1,L2GC2
      real*8 a,b,c,a2,b2,c2,k2
      integer n2

      a       = ao_1cyl
      b       = bo_c1
      c       = co_c2
      a2      = ao_1cyl**2
      b2      = bo_c1**2
      c2      = co_c2**2
      n2      = n**2
      k2      = k**2
      L2GC1   = lame_c1 + 2.0D0 * shear_c1
      L2GC2   = lame_c2 + 2.0D0 * shear_c2



C BOUNDARY CONDITION #1 (EQ 229)

      smrc2(1,1) = L2GC2*d2_SP_CY2_c1 + (lame_c2/c)*
     1d1_SP_CY2_c1 - lame_c2*SP_CY2_c1*(n2/c2 + k2) +
     1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY2_c1

      smrc2(1,2) = L2GC2*d2_SP_CY2_c2 + (lame_c2/c)*
     1d1_SP_CY2_c2 - lame_c2*SP_CY2_c2*(n2/c2 + k2) +
     1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY2_c2

      smrc2(1,3) = -2.0D0*shear_c2*n/c2*(VXP_CY2_c1-c*
     1d1_VXP_CY2_c1) +
     1n*ro*Om**2*OFSC*VXP_CY2_c1/(d1_OFSC*c)

      smrc2(1,4) = -2.0D0*shear_c2*n/c2*(VXP_CY2_c2-c*
     1d1_VXP_CY2_c2) +
     1n*ro*Om**2*OFSC*VXP_CY2_c2/(d1_OFSC*c)

      smrc2(1,5) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_c1
     1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY2_c1/d1_OFSC

      smrc2(1,6) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_c2
     1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY2_c2/d1_OFSC

      smrc2(1,7) = (0.0D0,0.0D0)
```

```
      smrc2(1,8) = (0.0D0,0.0D0)

      smrc2(1,9) = (0.0D0,0.0D0)

      smrc2(1,10)= (0.0D0,0.0D0)

      smrc2(1,11)= (0.0D0,0.0D0)

      smrc2(1,12)= (0.0D0,0.0D0)

      smrc2(1,13)= (0.0D0,0.0D0)



C BOUNDARY CONDITION #2 (EQ 215)

      smrc2(2,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_c1

      smrc2(2,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_c2

      smrc2(2,3) = (0.0D0,1.0D0)*k*n*shear_c2*VXP_CY2_c1/c

      smrc2(2,4) = (0.0D0,1.0D0)*k*n*shear_c2*VXP_CY2_c2/c

      smrc2(2,5) = shear_c2*(VRTP_CY2_c1*(n/c2 - k2 +
     1 1.0D0/c2) - d1_VRTP_CY2_c1*(n + 1.0D0)/c - d2_VRTP_CY2_c1)

      smrc2(2,6) = shear_c2*(VRTP_CY2_c2*(n/c2 - k2 +
     1 1.0D0/c2) - d1_VRTP_CY2_c2*(n + 1.0D0)/c - d2_VRTP_CY2_c2)

      smrc2(2,7) = (0.0D0,0.0D0)

      smrc2(2,8) = (0.0D0,0.0D0)

      smrc2(2,9) = (0.0D0,0.0D0)

      smrc2(2,10) = (0.0D0,0.0D0)

      smrc2(2,11) = (0.0D0,0.0D0)

      smrc2(2,12) = (0.0D0,0.0D0)

      smrc2(2,13) = (0.0D0,0.0D0)
```

```
C BOUNDARY CONDITION #3 (EQ 216)

      smrc2(3,1) = (shear_c2*2.0D0*n/c)*((1.0D0/c)*
     1SP_CY2_c1 - d1_SP_CY2_c1)

      smrc2(3,2) = (shear_c2*2.0D0*n/c)*((1.0D0/c)*
     1SP_CY2_c2 - d1_SP_CY2_c2)

      smrc2(3,3) = shear_c2*(-d2_VXP_CY2_c1 + 1.0D0/c*
     1d1_VXP_CY2_c1 - n2/c2*VXP_CY2_c1)

      smrc2(3,4) = shear_c2*(-d2_VXP_CY2_c2 + 1.0D0/c*
     1d1_VXP_CY2_c2 - n2/c2*VXP_CY2_c2)

      smrc2(3,5) = (0.0D0,1.0D0)*k*shear_c2*(d1_VRTP_CY2_c1
     1- VRTP_CY2_c1*(1.0D0 + n)/c)

      smrc2(3,6) = (0.0D0,1.0D0)*k*shear_c2*(d1_VRTP_CY2_c2
     1- VRTP_CY2_c2*(1.0D0 + n)/c)

      smrc2(3,7) = (0.0D0,0.0D0)

      smrc2(3,8) = (0.0D0,0.0D0)

      smrc2(3,9) = (0.0D0,0.0D0)

      smrc2(3,10) = (0.0D0,0.0D0)

      smrc2(3,11) = (0.0D0,0.0D0)

      smrc2(3,12) = (0.0D0,0.0D0)

      smrc2(3,13) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #5 (EQ 218)

      smrc2(4,1) = L2GC2*d2_SP_CY2_b1 + lame_c2/b
     1*d1_SP_CY2_b1 - lame_c2*SP_CY2_b1*(n2/b2 + k2)

      smrc2(4,2) = L2GC2*d2_SP_CY2_b2 + lame_c2/b
     1*d1_SP_CY2_b2 - lame_c2*SP_CY2_b2*(n2/b2 + k2)

      smrc2(4,3) = -2.0D0*shear_c2*n/b2*(VXP_CY2_b1 - b
     1*d1_VXP_CY2_b1)
```

```
      smrc2(4,4) = -2.0D0*shear_c2*n/b2*(VXP_CY2_b2 - b
     1*d1_VXP_CY2_b2)

      smrc2(4,5) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_b1

      smrc2(4,6) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_b2

      smrc2(4,7) = -L2GC1*d2_SP_CY1_b1 - lame_c1/b*
     1d1_SP_CY1_b1 + lame_c1*SP_CY1_b1*(n2/b2 + k2)

      smrc2(4,8) = -L2GC1*d2_SP_CY1_b2 - lame_c1/b*
     1d1_SP_CY1_b2 + lame_c1*SP_CY1_b2*(n2/b2 + k2)

      smrc2(4,9) = 2.0D0*shear_c1*n/b2*(VXP_CY1_b1 -
     1b*d1_VXP_CY1_b1)

      smrc2(4,10) = 2.0D0*shear_c1*n/b2*(VXP_CY1_b2 -
     1b*d1_VXP_CY1_b2)

      smrc2(4,11) = -(0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b1

      smrc2(4,12) = -(0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b2

      smrc2(4,13) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #6 (EQ 219)

      smrc2(5,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_b1

      smrc2(5,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_b2

      smrc2(5,3) = (0.0D0,1.0D0)*k*n*shear_c2/b*VXP_CY2_b1

      smrc2(5,4) = (0.0D0,1.0D0)*k*n*shear_c2/b*VXP_CY2_b2

      smrc2(5,5) = shear_c2*(VRTP_CY2_b1*((n+1.0D0)/b2 - k2)
     1- d1_VRTP_CY2_b1*(n+1.0D0)/b - d2_VRTP_CY2_b1)

      smrc2(5,6) = shear_c2*(VRTP_CY2_b2*((n+1.0D0)/b2 - k2)
     1- d1_VRTP_CY2_b2*(n+1.0D0)/b - d2_VRTP_CY2_b2)

      smrc2(5,7) = (0.0D0,-1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b1
```

```
      smrc2(5,8) = (0.0D0,-1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b2

      smrc2(5,9) = (0.0D0,-1.0D0)*k*n*shear_c1/b*VXP_CY1_b1

      smrc2(5,10) = (0.0D0,-1.0D0)*k*n*shear_c1/b*VXP_CY1_b2

      smrc2(5,11) = -(shear_c1*(VRTP_CY1_b1*(n/b2 - k2 + 1.0D0/b2) -
     1d1_VRTP_CY1_b1*(n/b+1.0D0/b) - d2_VRTP_CY1_b1))

      smrc2(5,12) = -(shear_c1*(VRTP_CY1_b2*(n/b2 - k2 + 1.0D0/b2) -
     1d1_VRTP_CY1_b2*(n/b+1.0D0/b) - d2_VRTP_CY1_b2))

      smrc2(5,13) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #7 (EQ 220)

      smrc2(6,1) = shear_c2*2.0D0*n/b*(1.0D0/b*SP_CY2_b1
     1- d1_SP_CY2_b1)

      smrc2(6,2) = shear_c2*2.0D0*n/b*(1.0D0/b*SP_CY2_b2
     1- d1_SP_CY2_b2)

      smrc2(6,3) = shear_c2*(-d2_VXP_CY2_b1 + d1_VXP_CY2_b1/b
     1 - n2/b2*VXP_CY2_b1)

      smrc2(6,4) = shear_c2*(-d2_VXP_CY2_b2 + d1_VXP_CY2_b2/b
     1 - n2/b2*VXP_CY2_b2)

      smrc2(6,5) = (0.0D0,1.0D0)*k*shear_c2*(d1_VRTP_CY2_b1 -
     1 VRTP_CY2_b1*(1.0D0+n)/b)

      smrc2(6,6) = (0.0D0,1.0D0)*k*shear_c2*(d1_VRTP_CY2_b2 -
     1 VRTP_CY2_b2*(1.0D0+n)/b)

      smrc2(6,7) = -2.0D0*n*shear_c1/b*(1.0D0/b*SP_CY1_b1
     1 - d1_SP_CY1_b1)

      smrc2(6,8) = -2.0D0*n*shear_c1/b*(1.0D0/b*SP_CY1_b2
     1 - d1_SP_CY1_b2)

      smrc2(6,9) = -shear_c1*(-d2_VXP_CY1_b1 + d1_VXP_CY1_b1/b
     1 - n2/b2*VXP_CY1_b1)

      smrc2(6,10) = -shear_c1*(-d2_VXP_CY1_b2 + d1_VXP_CY1_b2/b
```

```
     1 - n2/b2*VXP_CY1_b2)

          smrc2(6,11) = (0.0D0,-1.0D0)*k*shear_c1*(d1_VRTP_CY1_b1 -
     1VRTP_CY1_b1*(1.0/b + n/b))

          smrc2(6,12) = (0.0D0,-1.0D0)*k*shear_c1*(d1_VRTP_CY1_b2 -
     1VRTP_CY1_b2*(1.0D0/b + n/b))

          smrc2(6,13)= (0.0D0,0.0D0)



C BOUNDARY CONDITION #8 (EQ 221)

          smrc2(7,1) = d1_SP_cy2_b1

          smrc2(7,2) = d1_SP_cy2_b2

          smrc2(7,3) = n*VXP_CY2_b1/b

          smrc2(7,4) = n*VXP_CY2_b2/b

          smrc2(7,5) = (0.0D0,1.0D0)*k*VRTP_CY2_b1

          smrc2(7,6) = (0.0D0,1.0D0)*k*VRTP_CY2_b2

          smrc2(7,7) = -d1_SP_CY1_b1

          smrc2(7,8) = -d1_SP_CY1_b2

          smrc2(7,9) = -n*VXP_CY1_b1/b

          smrc2(7,10) = -n*VXP_CY1_b2/b

          smrc2(7,11) = (0.0D0,-1.0D0)*k*VRTP_CY1_b1

          smrc2(7,12) = (0.0D0,-1.0D0)*k*VRTP_CY1_b2

          smrc2(7,13) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #9 (EQ 222)

          smrc2(8,1) = -n*SP_CY2_b1/b

```

```
        smrc2(8,2) = -n*SP_CY2_b2/b

        smrc2(8,3) = -d1_VXP_CY2_b1

        smrc2(8,4) = -d1_VXP_CY2_b2

        smrc2(8,5) = (0.0D0,1.0D0)*k*VRTP_CY2_b1

        smrc2(8,6) = (0.0D0,1.0D0)*k*VRTP_CY2_b2

        smrc2(8,7) = n*SP_CY1_b1/b

        smrc2(8,8) = n/b*SP_CY1_b2

        smrc2(8,9) = d1_VXP_CY1_b1

        smrc2(8,10) = d1_VXP_CY1_b2

        smrc2(8,11)= (0.0D0,-1.0D0)*k*VRTP_CY1_b1

        smrc2(8,12)= (0.0D0,-1.0D0)*k*VRTP_CY1_b2

        smrc2(8,13)= (0.0D0,0.0D0)



C BOUNDARY CONDITION #10 (EQ 223)

        smrc2(9,1) = (0.0D0,1.0D0)*k*SP_CY2_b1

        smrc2(9,2) = (0.0D0,1.0D0)*k*SP_CY2_b2

        smrc2(9,3) = (0.0D0,0.0D0)

        smrc2(9,4) = (0.0D0,0.0D0)

        smrc2(9,5) = -d1_VRTP_CY2_b1 - (n + 1.0D0)*VRTP_CY2_b1
   1/b

        smrc2(9,6) = -d1_VRTP_CY2_b2 - (n + 1.0D0)*VRTP_CY2_b2
   1/b

        smrc2(9,7) = (0.0D0,-1.0D0)*k*SP_CY1_b1

        smrc2(9,8) = (0.0D0,-1.0D0)*k*SP_CY1_b2
```

```
      smrc2(9,9) = (0.0D0,0.0D0)

      smrc2(9,10) = (0.0D0,0.0D0)

      smrc2(9,11) = d1_VRTP_CY1_b1 + VRTP_CY1_b1*(n+1.0D0)/b

      smrc2(9,12) = d1_VRTP_CY1_b2 + VRTP_CY1_b2*(n+1.0D0)/b

      smrc2(9,13) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #11 (EQ 224)


      smrc2(10,1) = (0.0D0,0.0D0)

      smrc2(10,2) = (0.0D0,0.0D0)

      smrc2(10,3) = (0.0D0,0.0D0)

      smrc2(10,4) = (0.0D0,0.0D0)

      smrc2(10,5) = (0.0D0,0.0D0)

      smrc2(10,6) = (0.0D0,0.0D0)

      smrc2(10,7) = L2GC1*d2_SP_CY1_a1 + (lame_c1/a)*
     1d1_SP_CY1_a1 - lame_c1*SP_CY1_a1*(n2/a2 + k2)

      smrc2(10,8) = L2GC1*d2_SP_CY1_a2 + (lame_c1/a)*
     1d1_SP_CY1_a2 - lame_c1*SP_CY1_a2*(n2/a2 + k2)

      smrc2(10,9) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a1
     1- a*d1_VXP_CY1_a1)

      smrc2(10,10) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a2
     1- a*d1_VXP_CY1_a2)

      smrc2(10,11) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a1

      smrc2(10,12) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a2

      smrc2(10,13) = +ri*Om**2*IFSC
```

```
C BOUNDARY CONDITION #12 (EQ 225)


        smrc2(11,1) = (0.0D0,0.0D0)

        smrc2(11,2) = (0.0D0,0.0D0)

        smrc2(11,3) = (0.0D0,0.0D0)

        smrc2(11,4) = (0.0D0,0.0D0)

        smrc2(11,5) = (0.0D0,0.0D0)

        smrc2(11,6) = (0.0D0,0.0D0)

        smrc2(11,7) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a1

        smrc2(11,8) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a2

        smrc2(11,9) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a1/a

        smrc2(11,10) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a2/a

        smrc2(11,11) = shear_c1*(VRTP_CY1_a1*(n/a2 - k2 +
     1 1.0D0/a2) - d1_VRTP_CY1_a1*(n + 1.0D0)/a - d2_VRTP_CY1_a1)

        smrc2(11,12) = shear_c1*(VRTP_CY1_a2*(n/a2 - k2 +
     1 1.0D0/a2) - d1_VRTP_CY1_a2*(n + 1.0D0)/a - d2_VRTP_CY1_a2)

        smrc2(11,13) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #13 (EQ 226)


        smrc2(12,1) = (0.0D0,0.0D0)

        smrc2(12,2) = (0.0D0,0.0D0)

        smrc2(12,3) = (0.0D0,0.0D0)

        smrc2(12,4) = (0.0D0,0.0D0)

        smrc2(12,5) = (0.0D0,0.0D0)
```

```
      smrc2(12,6) = (0.0D0,0.0D0)

      smrc2(12,7) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
     1SP_CY1_a1 - d1_SP_CY1_a1)

      smrc2(12,8) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
     1SP_CY1_a2 - d1_SP_CY1_a2)

      smrc2(12,9) = shear_c1*(-d2_VXP_CY1_a1 + 1.0D0/a*
     1d1_VXP_CY1_a1 - n2/a2*VXP_CY1_a1)

      smrc2(12,10) = shear_c1*(-d2_VXP_CY1_a2 + 1.0D0/a*
     1d1_VXP_CY1_a2 - n2/a2*VXP_CY1_a2)

      smrc2(12,11) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a1
     1- VRTP_CY1_a1*(1.0D0 + n)/a)

      smrc2(12,12) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a2
     1- VRTP_CY1_a2*(1.0D0 + n)/a)


      smrc2(12,13) = (0.0D0,0.0D0)


C BOUNDARY CONDITION #14 (EQ 227)


      smrc2(13,1) = (0.0D0,0.0D0)

      smrc2(13,2) = (0.0D0,0.0D0)

      smrc2(13,3) = (0.0D0,0.0D0)

      smrc2(13,4) = (0.0D0,0.0D0)

      smrc2(13,5) = (0.0D0,0.0D0)

      smrc2(13,6) = (0.0D0,0.0D0)

      smrc2(13,7) = d1_SP_CY1_a1

      smrc2(13,8) = d1_SP_CY1_a2

      smrc2(13,9) = n/a*VXP_CY1_a1
```

```
      smrc2(13,10) = n/a*VXP_CY1_a2

      smrc2(13,11) = (0.0D0,1.0D0)*k*VRTP_CY1_a1

      smrc2(13,12) = (0.0D0,1.0D0)*k*VRTP_CY1_a2

      smrc2(13,13) = -d1_IFSC


C END BOUNDARY CONDITIONS FOR MATRIX "smrc2"



      return
      end




      SUBROUTINE ABC_RC2_INVERT(n,exctype,smrc2,a,c)


C EXTERNAL VARIABLES

      integer n,exctype
      real*8 a,c
      complex*16 smrc2(13,13)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
```

```
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1       SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1       SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1       SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1       VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1       VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1       VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1       VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1       VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1       VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1       VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1       VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1       lame_c1,shear_c1,cl_c1,ct_c1,
     1       A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C *******************************************************************


C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2


      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
     1       SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
     1       SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
     1       SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
```

```
     1                VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
     1                VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
     1                VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
     1                VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
     1                VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
     1                VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
     1                VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
     1                VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
     1                lame_c2,shear_c2,cl_c2,ct_c2,
     1                A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C *******************************************************************



C DEFINITIONS FOR COMMON BLOCK /ROD/

        common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


        complex*16 SP_rod,d1_SP_rod,d2_SP_rod
        complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
        complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
        complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C *******************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        complex*16 OFSC,d1_OFSC,M_OF

        common /OFLUID/ OFSC,d1_OFSC,M_OF

C *******************************************************************


C *******************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

        common /IFLUID/ IFSC,d1_IFSC,D_IF

        complex*16 IFSC,d1_IFSC,D_IF

```

```
C *******************************************************************


C INTERNAL VARIABLES

        integer size,iflag
        complex*16 smrc2inv(13,13),workrc2(13,26)

        size = 13
        iflag = 0


        CALL MINV(smrc2,smrc2inv,workrc2,size,iflag)



C RADIAL EXCITATION exctype = 1
C AXIAL  EXCITATION exctype = 0

        if (exctype .eq. 1) then
           A1_C2 = -smrc2inv(1,1)
           A2_C2 = -smrc2inv(2,1)
           B1_C2 = -smrc2inv(3,1)
           B2_C2 = -smrc2inv(4,1)
           C1_C2 = -smrc2inv(5,1)
           C2_C2 = -smrc2inv(6,1)
           A1_C1 = -smrc2inv(7,1)
           A2_C1 = -smrc2inv(8,1)
           B1_C1 = -smrc2inv(9,1)
           B2_C1 = -smrc2inv(10,1)
           C1_C1 = -smrc2inv(11,1)
           C2_C1 = -smrc2inv(12,1)
           D_IF = -smrc2inv(13,1)
           M_OF =(A1_C2*d1_SP_CY2_c1 +
     1       A2_C2*d1_SP_CY2_c2  +
     1       B1_C2*n/c*VXP_CY2_c1 +
     1       B2_C2*n/c*VXP_CY2_c2 +
     1       C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
     1       C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

        elseif (exctype .eq. 0) then
           A1_C2 = -smrc2inv(1,2)
           A2_C2 = -smrc2inv(2,2)
           B1_C2 = -smrc2inv(3,2)
           B2_C2 = -smrc2inv(4,2)
           C1_C2 = -smrc2inv(5,2)
```

```
            C2_C2 = -smrc2inv(6,2)
            A1_C1 = -smrc2inv(7,2)
            A2_C1 = -smrc2inv(8,2)
            B1_C1 = -smrc2inv(9,2)
            B2_C1 = -smrc2inv(10,2)
            C1_C1 = -smrc2inv(11,2)
            C2_C1 = -smrc2inv(12,2)
            D_IF = -smrc2inv(13,2)
            M_OF =(A1_C2*d1_SP_CY2_c1 +
     1       A2_C2*d1_SP_CY2_c2 +
     1       B1_C2*n/c*VXP_CY2_c1 +
     1       B2_C2*n/c*VXP_CY2_c2 +
     1       C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
     1       C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

        endif

        return
        end

        SUBROUTINE OUTPUT_RC2(tft,n,k,r,value)

C EXTERNAL VARIABLES

        integer n,tft
        real*8 k,r,zero
        complex*16 value

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

        complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
        complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
        omplex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
        complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

        complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
        complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
        complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
        complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
```

```
      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1          SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1          SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1          SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1          VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1          VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1          VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1          VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1          VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1          VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1          VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1          VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1          lame_c1,shear_c1,cl_c1,ct_c1,
     1          A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C ********************************************************************


C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
```

```
      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
     1          SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
     1          SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
     1          SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
     1          VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
     1          VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
     1          VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
     1          VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
     1          VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
     1          VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
     1          VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
     1          VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
     1          lame_c2,shear_c2,cl_c2,ct_c2,
     1          A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C ***********************************************************************


C DEFINITIONS FOR COMMON BLOCK /ROD/

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ***********************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      complex*16 OFSC,d1_OFSC,M_OF

      common /OFLUID/ OFSC,d1_OFSC,M_OF

C ***********************************************************************


C ***********************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/
```

```
      common /IFLUID/ IFSC,d1_IFSC,D_IF

      complex*16 IFSC,d1_IFSC,D_IF

C ********************************************************************


C INTERNAL VARIABLES

      integer n2
      real*8 k2,r2,no,p11,p12
      complex*16 dpp,exx,err,Srr,Sxx,ett,uc,vc,wc
      complex*16 A1C2,A2C2,B1C2,B2C2,C1C2,C2C2
      complex*16 L2GC1,L2GC2

      n2 = n**2
      r2 = r**2
      k2 = k**2

      zero = 1.000D-30
      no   = 1.460D0
      p11  = 0.126D0
      p12  = 0.270D0
      L2GC1 = lame_c1 + 2.0D0*shear_c1
      L2GC2 = lame_c2 + 2.0D0*shear_c2


C  RADIAL STRESS/(Pr or Px)                                 tft = 0
C  LONGITUDINAL STRESS                                      tft = 1
C  AXIAL  DISPLACEMENT                                      tft = 2
C  THETA  DISPLACEMENT                                      tft = 3
C  RADIAL DISPLACEMENT                                      tft = 4
C  LONGITUDINAL STRAIN     e11/(Pr or Px)                   tft = 5
C  THETA  STRAIN           ett/(Pr or Px)                   tft = 6
C  RADIAL STRAIN           err/(Pr or Px)                   tft = 7
C                                                           tft = 8
C  OPTIC exx=0@k=0         ((dp/p)(r))/(Pr or Px)           tft = 9
C  OPTIC exx=const@k=0     ((dp/p)(r))/(Pr or Px)           tft = 10






      if (tft .eq. 0) then
```

```
      A1C2 = A1_C2*(L2GC2*d2_SP_CY2_b1 + lame_c2*
     1 d1_SP_CY2_b1/r - lame_c2*SP_CY2_b1*(n2/
     1 r2 + k2))

      A2C2 = A2_C2*(L2GC2*d2_SP_CY2_b2 + lame_c2*
     1 d1_SP_CY2_b2/r - lame_c2* SP_CY2_b2*(n2/
     1 r2 + k2))

      B1C2 = B1_C2*-2.0D0*shear_c2*n/r**2*(VXP_CY2_b1 -
     1 r*d1_VXP_CY2_b1)

      B2C2 = B2_C2*-2.0D0*shear_c2*n/r**2*(VXP_CY2_b2 -
     1 r*d1_VXP_CY2_b2)

      C1C2 = (0.0D0,1.0D0)*C1_C2*2.0D0*shear_c2*k*d1_VRTP_CY2_b1

      C2C2 = (0.0D0,1.0D0)*C2_C2*2.0D0*shear_c2*k*d1_VRTP_CY2_b2

      Srr = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

      value = Srr


C LONGITUDINAL STRESS USING (EQ 153)
    elseif(tft .eq. 1) then

      A1C2 = A1_C2*(lame_c2*(d2_SP_CY2_b1 + d1_SP_CY2_b1/r)
     1   - SP_CY2_b1*(lame_c2*n2/r2 + k2*(lame_c2
     1   + 2.0D0*shear_c2)))

      A2C2 = A2_C2*(lame_c2*(d2_SP_CY2_b2 + d1_SP_CY2_b2/r)
     1   - SP_CY2_b2*(lame_c2*n2/r2 + k2*(lame_c2
     1   + 2.0D0*shear_c2)))

      C1C2 = C1_C2*((0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b1 +
     1   VRTP_CY2_b1*(n + 1.0D0)/r))*2.0D0*shear_c2

      C2C2 = C2_C2*((0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b2 +
     1   VRTP_CY2_b2*(n + 1.0D0)/r))*2.0D0*shear_c2

      Sxx = A1C2 + A2C2 + C1C2 + C2C2

      value = Sxx
```

```
C AXIAL DISPLACEMENT USING (EQ 149)
          elseif(tft .eq. 2) then

            A1C2 = A1_C2*(0.0D0,1.0D0)*k*SP_CY2_b1

            A2C2 = A2_C2*(0.0D0,1.0D0)*k*SP_CY2_b2

            C1C2 = C1_C2*(-d1_VRTP_CY2_b1 - VRTP_CY2_b1*
     1        (n + 1.0D0)/r)

            C2C2 = C2_C2*(-d1_VRTP_CY2_b2 - VRTP_CY2_b2*
     1        (n + 1.0D0)/r)

            wc = A1C2 + A2C2 + C1C2 + C2C2

            value = wc

C THETA DISPLACEMENT USING (EQ 150)
          elseif(tft .eq. 3) then


            A1C2 = A1_C2*-SP_CY2_b1*n/r

            A2C2 = A2_C2*-SP_CY2_b2*n/r

            B1C2 = -B1_C2*d1_VXP_CY2_b1

            B2C2 = -B2_C2*d1_VXP_CY2_b2

            C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1

            C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2

            vc   = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

            value = vc

C RADIAL DISPLACEMENT USING (EQ 148)
          elseif(tft .eq. 4) then

            A1C2 = A1_C2*d1_SP_CY2_b1

            A2C2 = A2_C2*d1_SP_CY2_b2

            B1C2 = B1_C2*VXP_CY2_b1*n/r

```

```
          B2C2 = B2_C2*VXP_CY2_b2*n/r

          C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1

          C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2

          uc   = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

          value = uc

C LONGITUDINAL STRAIN USING (EQ 147)
        elseif(tft .eq. 5) then

          A1C2 = -A1_C2*k2*SP_CY2_b1

          A2C2 = -A2_C2*k2_SP_CY2_b2

          C1C2 = C1_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b1 +
     1      VRTP_CY2_b1*(n + 1.0D0)/r)

          C2C2 = C2_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b2 +
     1      VRTP_CY2_b2*(n + 1.0D0)/r)

          exx  = A1C2 + A2C2 + C1C2 + C2C2

            if (zabs(exx) .lt. dabs(zero)) then
              value = zero
            endif

          value = exx

C THETA STRAIN USING (EQ 145)
        elseif(tft .eq. 6) then

          A1C2 = A1_C2*(d1_SP_CY2_b1/r - SP_CY2_b1*n2/r2)

          A2C2 = A2_C2*(d1_SP_CY2_b2/r - SP_CY2_b2*n2/r2)

          B1C2 = B1_C2*(-d1_VXP_CY2_b1*n/r + VXP_CY2_b1*n/r2)

          B2C2 = B2_C2*(-d1_VXP_CY2_b2*n/r + VXP_CY2_b2*n/r2)

          C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1*(n + 1.0D0)/r

          C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2*(n + 1.0D0)/r

```

```
      ett  = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

        if (zabs(ett) .lt. dabs(zero)) then
          value = zero
        endif

      value = ett

C RADIAL STRAIN USING (eq 143)
      elseif(tft .eq. 7) then

        A1C2 = A1_C2*d2_SP_CY2_b1

        A2C2 = A2_C2*d2_SP_CY2_b2

        B1C2 = B1_C2*(d1_VXP_CY2_b1 - VXP_CY2_b1/r)*n/r

        B2C2 = B2_C2*(d1_VXP_CY2_b2 - VXP_CY2_b2/r)*n/r

        C1C2 = C1_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b1

        C2C2 = C2_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b2

        err  = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

        if (zabs(err) .lt. dabs(zero)) then
          value = zero
        endif

      value = err

C UNUSED AT PRESENT
      elseif(tft .eq. 8) then
        value = 1.0D0

C OPTIC dP/P/uPa USING (EQ 147 & 143)
C exx = 0 @ k = 0
      elseif(tft .eq. 9) then

        A1C2 = -A1_C2*k2*SP_CY2_b1

        A2C2 = -A2_C2*k2_SP_CY2_b2

        C1C2 = C1_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b1 +
     1      VRTP_CY2_b1*(n + 1.0D0)/r)

```

```
      C2C2 = C2_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b2 +
     1   VRTP_CY2_b2*(n + 1.0D0)/r)

      exx  = A1C2 + A2C2 + C1C2 + C2C2

        if (zabs(exx) .lt. dabs(zero)) then
          exx = zero
        endif

      A1C2 = A1_C2*d2_SP_CY2_b1

      A2C2 = A2_C2*d2_SP_CY2_b2

      B1C2 = B1_C2*(d1_VXP_CY2_b1 - VXP_CY2_b1/r)*n/r

      B2C2 = B2_C2*(d1_VXP_CY2_b2 - VXP_CY2_b2/r)*n/r

      C1C2 = C1_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b1

      C2C2 = C2_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b2

      err  = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

        if (zabs(err) .lt. dabs(zero)) then
          err = zero
        endif

      dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)

      value = dpp

C OPTIC dp/p/uPa USING (EQ 145 & 143)
C exx = constant @ k = 0
    elseif(tft .eq. 10) then

      A1C2 = A1_C2*(d1_SP_CY2_b1/r - SP_CY2_b1*n2/r2)

      A2C2 = A2_C2*(d1_SP_CY2_b2/r - SP_CY2_b2*n2/r2)

      B1C2 = B1_C2*(-d1_VXP_CY2_b1*n/r + VXP_CY2_b1*n/r2)

      B2C2 = B2_C2*(-d1_VXP_CY2_b2*n/r + VXP_CY2_b2*n/r2)

      C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1*(n + 1.0D0)/r

      C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2*(n + 1.0D0)/r
```

```
       ett  = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

         if (zabs(ett) .lt. dabs(zero)) then
           value = zero
         endif

       A1C2 = A1_C2*d2_SP_CY2_b1

       A2C2 = A2_C2*d2_SP_CY2_b2

       B1C2 = B1_C2*(d1_VXP_CY2_b1 - VXP_CY2_b1/r)*n/r

       B2C2 = B2_C2*(d1_VXP_CY2_b2 - VXP_CY2_b2/r)*n/r

       C1C2 = C1_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b1

       C2C2 = C2_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b2

       err  = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2

         if (zabs(err) .lt. dabs(zero)) then
           err = zero
         endif

       exx = -(lame_rod/(lame_rod + 2.0D0 * shear_rod))*(ett + err)
       dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
       value = dpp

     endif



     return
     end
```

LISTING FOR fluids.f

fluids.f Sat Jun 10 14:41:58 1995

```
C ********************************************************************
C
C                 subprogram "fluids.f"
C
C This subprogram was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C ********************************************************************


      SUBROUTINE IFL_POT(n,r,k,Om,ci)

C EXTERNAL VARIABLES

      integer n
      real*8 r,k,Om,ci

C ********************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

      complex*16 IFSC,d1_IFSC,D_IF

      common /IFLUID/ IFSC,d1_IFSC,D_IF


C ********************************************************************

      double complex cbessj,d1cbessj


C INTERNAL VARIABLES

      complex*16 g1,arg

      arg = DCMPLX(((Om/ci)**2 - k**2), 0.0D0)
      g1  = zsqrt(arg)

        d1_IFSC = d1cbessj(n,g1,r)
           IFSC =   cbessj(n,g1,r)

```

```
      return
      end




      SUBROUTINE OFL_POT(n,r,k,Om,co)

      integer n
      real*8 r,k,Om,co

C ***********************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      common /OFLUID/ OFSC,d1_OFSC,M_OF

      complex*16 OFSC,d1_OFSC,M_OF

C ***********************************************************************

      complex*16 g2,f2
      double complex cbessh1,d1cbessh1,cbessk,d1cbessk

      if (dabs(k) .le. dabs(Om/co)) then

        g2 = dsqrt((Om/co)**2 - k**2)
            d1_OFSC = d1cbessh1(n,g2,r)
                OFSC =   cbessh1(n,g2,r)

      else

        f2 = dsqrt(k**2 - (Om/co)**2)
            d1_OFSC = d1cbessk(n,f2,r)
                OFSC =   cbessk(n,f2,r)

      endif

      return
      end
```

```
      SUBROUTINE OUTPUT_IF(tft,n,k,r,value,Om,ri)

C EXTERNAL VARIABLES

         integer n,tft
         real*8 k,r,Om,ri
         complex*16 value
C ***********************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

         complex*16 IFSC,d1_IFSC,D_IF

         common /IFLUID/ IFSC,d1_IFSC,D_IF


C ***********************************************************************

C INTERNAL VARIABLES

         complex*16 pi,vui

C OUTER FLUID PRESSURE TRANSFER FUNCTION USING (EQ F.14)
         if (tft .eq. 0)then

           pi = Om**2*ri*D_IF*IFSC

           value = pi

C OUTER FLUID VELOCITY USING (EQ F.16)
         elseif(tft .eq. 8)then

           vui = (0.0D0,-1.0D0)*Om*D_IF*d1_IFSC

           value = vui

         endif

         return
         end
```

```
      SUBROUTINE OUTPUT_OF(tft,n,k,r,value,Om,ro)

C EXTERNAL VARIABLES

      integer n,tft
      real*8 k,r,Om,ro
      complex*16 value
C ***********************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      common /OFLUID/ OFSC,d1_OFSC,M_OF

      complex*16 OFSC,d1_OFSC,M_OF

C ***********************************************************************

C INTERNAL VARIABLES

      complex*16 ps,vws

C OUTER FLUID PRESSURE TRANSFER FUNCTION USING (EQ F.7)
      if (tft .eq. 0)then

        ps = Om**2*ro*M_OF*OFSC

        value = ps

C OUTER FLUID VELOCITY USING (EQ F.8a)
      elseif(tft .eq. 8)then

        vws = (0.0D0,-1.0D0)*Om*M_OF*d1_OFSC

        value = vws

      endif

      return
      end
```

LISTING FOR smc1.f

```
C **********************************************************************
C
C              subprogram "smc1.f"
C
C This subprogram was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C **********************************************************************


      SUBROUTINE SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,
     1Om,ro,co,ri,ci,smc1)

C EXTERNAL VARIABLES

      integer n
      real*8 k,ao_1cyl,bo_c1,Om,ro,co,ri,ci
      complex*16 smc1(7,7)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1

      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1        SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1        SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
```

```
     1              SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1              VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1              VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1              VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1              VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1              VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1              VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1              VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1              VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1              lame_c1,shear_c1,cl_c1,ct_c1,
     1              A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C *****************************************************************


C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      common /OFLUID/ OFSC,d1_OFSC,M_OF

      complex*16 OFSC,d1_OFSC,M_OF

C *****************************************************************


C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

      common /IFLUID/ IFSC,d1_IFSC,D_IF

      complex*16 IFSC,d1_IFSC,D_IF

C *****************************************************************


C INTERNAL VARIABLES

      complex*16 L2GC1
      real*8 a,b,b2,k2,a2
      integer n2

      a    = ao_1cyl
      b    = bo_c1
      a2   = ao_1cyl**2
```

```
      b2    = bo_c1**2
      n2    = n**2
      k2    = k**2
      L2GC1  = lame_c1 + 2.0D0 * shear_c1


C BOUNDARY CONDITION #1 (EQ 113)

      smc1(1,1) = L2GC1*d2_SP_CY1_b1 + (lame_c1/bo_c1)*
     1d1_SP_CY1_b1 - lame_c1*SP_CY1_b1*(n2/b2 + k2) +
     1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY1_b1

      smc1(1,2) = L2GC1*d2_SP_CY1_b2 + (lame_c1/bo_c1)*
     1d1_SP_CY1_b2 - lame_c1*SP_CY1_b2*(n2/b2 + k2) +
     1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY1_b2

      smc1(1,3) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b1-bo_c1*
     1d1_VXP_CY1_b1) +
     1n*ro*Om**2*OFSC*VXP_CY1_b1/(d1_OFSC*b)

      smc1(1,4) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b2-bo_c1*
     1d1_VXP_CY1_b2) +
     1n*ro*Om**2*OFSC*VXP_CY1_b2/(d1_OFSC*b)

      smc1(1,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b1
     1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY1_b1/d1_OFSC

      smc1(1,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b2
     1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY1_b2/d1_OFSC

      smc1(1,7) = (0.0D0,0.0D0)



C BOUNDARY CONDITION #2 (EQ 116)

      smc1(2,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b1

      smc1(2,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b2

      smc1(2,3) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b1/bo_c1

      smc1(2,4) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b2/bo_c1

      smc1(2,5) = shear_c1*(VRTP_CY1_b1*(n/b2 - k2 +
     1 1.0D0/b2) - d1_VRTP_CY1_b1*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b1)
```

```
      smc1(2,6) = shear_c1*(VRTP_CY1_b2*(n/b2 - k2 +
     1 1.0D0/b2) - d1_VRTP_CY1_b2*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b2)

      smc1(2,7) = (0.0D0,0.0D0)




C BOUNDARY CONDITION #3 (EQ 120)

      smc1(3,1) = (shear_c1*2.0D0*n/bo_c1)*((1.0D0/bo_c1)*
     1SP_CY1_b1 - d1_SP_CY1_b1)

      smc1(3,2) = (shear_c1*2.0D0*n/bo_c1)*((1.0D0/bo_c1)*
     1SP_CY1_b2 - d1_SP_CY1_b2)

      smc1(3,3) = shear_c1*(-d2_VXP_CY1_b1 + 1.0D0/bo_c1*
     1d1_VXP_CY1_b1 - n2/b2*VXP_CY1_b1)

      smc1(3,4) = shear_c1*(-d2_VXP_CY1_b2 + 1.0D0/bo_c1*
     1d1_VXP_CY1_b2 - n2/b2*VXP_CY1_b2)

      smc1(3,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_b1
     1- VRTP_CY1_b1*(1.0D0 + n)/bo_c1)

      smc1(3,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_b2
     1- VRTP_CY1_b2*(1.0D0 + n)/bo_c1)

      smc1(3,7) = (0.0D0,0.0D0)




C BOUNDARY CONDITION #11 (EQ 224)




      smc1(4,1) = L2GC1*d2_SP_CY1_a1 + (lame_c1/a)*
     1d1_SP_CY1_a1 - lame_c1*SP_CY1_a1*(n2/a2 + k2)

      smc1(4,2) = L2GC1*d2_SP_CY1_a2 + (lame_c1/a)*
     1d1_SP_CY1_a2 - lame_c1*SP_CY1_a2*(n2/a2 + k2)

      smc1(4,3) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a1-a*
```

```
     1d1_VXP_CY1_a1)

         smc1(4,4) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a2-a*
     1d1_VXP_CY1_a2)

         smc1(4,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a1

         smc1(4,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a2

         smc1(4,7) = + ri*Om**2*IFSC


C BOUNDARY CONDITION #12 (EQ 225)



         smc1(5,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a1

         smc1(5,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a2

         smc1(5,3) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a1/a

         smc1(5,4) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a2/a

         smc1(5,5) = shear_c1*(VRTP_CY1_a1*((n+1.0D0)/a2 - k2)
     1- d1_VRTP_CY1_a1*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a1)

         smc1(5,6) = shear_c1*(VRTP_CY1_a2*((n+1.0D0)/a2 - k2)
     1- d1_VRTP_CY1_a2*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a2)

         smc1(5,7) = (0.0D0,0.0D0)





C BOUNDARY CONDITION #13 (EQ 226)



         smc1(6,1) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
     1SP_CY1_a1 - d1_SP_CY1_a1)

         smc1(6,2) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
     1SP_CY1_a2 - d1_SP_CY1_a2)

```

```
      smc1(6,3) = shear_c1*(-d2_VXP_CY1_a1 + 1.0D0/a*
     1d1_VXP_CY1_a1 - n2/a2*VXP_CY1_a1)

      smc1(6,4) = shear_c1*(-d2_VXP_CY1_a2 + 1.0D0/a*
     1d1_VXP_CY1_a2 - n2/a2*VXP_CY1_a2)

      smc1(6,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a1
     1- VRTP_CY1_a1*(1.0D0 + n)/a)

      smc1(6,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a2
     1- VRTP_CY1_a2*(1.0D0 + n)/a)

      smc1(6,7) = (0.0D0,0.0D0)




C BOUNDARY CONDITION #14 (EQ 227)



      smc1(7,1) = d1_SP_CY1_a1

      smc1(7,2) = d1_SP_CY1_a2

      smc1(7,3) = n/a*VXP_CY1_a1

      smc1(7,4) = n/a*VXP_CY1_a2

      smc1(7,5) = (0.0D0,1.0D0)*k*VRTP_CY1_a1

      smc1(7,6) = (0.0D0,1.0D0)*k*VRTP_CY1_a2

      smc1(7,7) = -d1_IFSC



C END BOUNDARY CONDITIONS FOR MATRIX "smc1"



      return
      end
```

```
      SUBROUTINE ABC_C1_INVERT(n,exctype,smc1,a,b)


C EXTERNAL VARIABLES

      integer n,exctype
      real*8 a,b
      complex*16 smc1(7,7)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1           SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1           SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1           SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1           VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1           VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1           VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1           VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1           VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1           VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1           VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1           VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1           lame_c1,shear_c1,cl_c1,ct_c1,
     1           A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
```

```
C ****************************************************************


C ****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

        complex*16 OFSC,d1_OFSC,M_OF

        common /OFLUID/ OFSC,d1_OFSC,M_OF

C ****************************************************************


C ****************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

        common /IFLUID/ IFSC,d1_IFSC,D_IF

        complex*16 IFSC,d1_IFSC,D_IF

C ****************************************************************


C INTERNAL VARIABLES

        integer size,iflag
        complex*16 smc1inv(7,7),workc1(7,14)

        size = 7
        iflag = 0


        CALL MINV(smc1,smc1inv,workc1,size,iflag)





C RADIAL EXCITATION exctype = 1
C AXIAL  EXCITATION exctype = 0

        if (exctype .eq. 1) then
          A1_C1 = -smc1inv(1,1)
          A2_C1 = -smc1inv(2,1)
```

```
      B1_C1 = -smc1inv(3,1)
      B2_C1 = -smc1inv(4,1)
      C1_C1 = -smc1inv(5,1)
      C2_C1 = -smc1inv(6,1)
      M_OF = (A1_C1*d1_SP_CY1_b1 +
1          A2_C1*d1_SP_CY1_b2 +
1          B1_C1*n/b*VXP_CY1_b1 +
1          B2_C1*n/b*VXP_CY1_b2 +
1          C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
1          C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
      D_IF = -smc1inv(7,1)
   elseif (exctype .eq. 0) then
      A1_C1 = -smc1inv(1,2)
      A2_C1 = -smc1inv(2,2)
      B1_C1 = -smc1inv(3,2)
      B2_C1 = -smc1inv(4,2)
      C1_C1 = -smc1inv(5,2)
      C2_C1 = -smc1inv(6,2)
      M_OF = (A1_C1*d1_SP_CY1_b1 +
1          A2_C1*d1_SP_CY1_b2 +
1          B1_C1*n/b*VXP_CY1_b1 +
1          B2_C1*n/b*VXP_CY1_b2 +
1          C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
1          C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
      D_IF = -smc1inv(7,2)
   endif



   return
   end
```

CROSS REFERENCE

SUBPROGRAMS

Cross Reference   Thu Oct 19 16:48:44 1995
Subprograms :

```
Symbol      File/Subprogram       Line
------      ---- ----------       ----

abc_c1_invert
            mr2cf.f/MAIN            316
            smc1.f/abc_c1_invert    279D
abc_rc1_invert
            c1.f/abc_rc1_invert     533D
            mr2cf.f/MAIN            388
abc_rc2_invert
            c2.f/abc_rc2_invert     800D
            mr2cf.f/MAIN            433
abc_rod_invert
              "  /  "               356
            rf.f/abc_rod_invert     201D
abc_rod_solve
             " /abc_rod_solve       152D
c1a_pot      c1.f/c1a_pot            14D
            mr2cf.f/MAIN            310, 326, 382, 397, 421, 445
c1b_pot      c1.f/c1b_pot            100D
            mr2cf.f/MAIN            311, 383, 422
c2b_pot      c2.f/c2b_pot            17D
            mr2cf.f/MAIN            423, 450
c2c_pot      c2.f/c2c_pot            107D
            mr2cf.f/MAIN            424
cbessh1      cbessl.f/cbessh1        483D
            fluids.f/ofl_pot        74
cbessh2      cbessl.f/cbessh2        497D
cbessi        "  /cbessi             141D
              "  /cbessk            273
              "  /d1cbessi          460, 460
cbessj       c1.f/c1a_pot            74, 82, 89
             " /c1b_pot             160, 169, 176
            c2.f/c2b_pot             78, 86, 93
             " /c2c_pot             170, 179, 186
            cbessl.f/cbessh1         489
              "  /cbessh2           503
              "  /cbessj            58D
              "  /cbessy            201
              "  /d1cbessj          330, 330
              "  /d2cbessj          346, 347, 347
```

Subprograms :

Symbol	File/Subprogram	Line
	fluids.f/ifl_pot	43
	rf.f/rod_pot	44, 50, 54
cbessk	cbessl.f/cbessk	244D
	" /d1cbessk	300, 300
	fluids.f/ofl_pot	80
cbessy	c1.f/c1a_pot	77, 85, 92
	" /c1b_pot	163, 172, 179
	c2.f/c2b_pot	81, 89, 96
	" /c2c_pot	173, 182, 189
	cbessl.f/cbessh1	489
	" /cbessh2	503
	" /cbessy	167D
	" /d1cbessy	367, 371, 371, 372, 376, 376, 377, 378, 384, 384, 388, 388, 389
	" /d2cbessy	413, 413, 414, 418, 418, 419, 420, 425, 425, 426, 427, 428, 429, 435, 435, 436, 440, 440, 441, 441
cdabs	" /cbessj	72
	" /cbessy	180
	mr2cf.f/MAIN	472
cdcos	cbessl.f/cbessj	126
	" /cbessy	233
cdlog	cbessl.f/cbessk	273
	" /cbessy	201
cdsin	" /cbessj	127
	" /cbessy	232
cdsqrt	" /cbessj	126
	" /cbessy	232
d1cbessh1	" /d1cbessh1	511D
	fluids.f/ofl_pot	73
d1cbessh2	cbessl.f/d1cbessh2	526D
d1cbessi	" /d1cbessi	453D
d1cbessj	c1.f/c1a_pot	73, 81, 88
	" /c1b_pot	159, 168, 175
	c2.f/c2b_pot	77, 85, 92
	" /c2c_pot	169, 178, 185
	cbessl.f/d1cbessh1	517
	" /d1cbessh2	532
	" /d1cbessj	323D
	fluids.f/ifl_pot	42
	rf.f/rod_pot	43, 49, 53

Subprograms :

```
Symbol      File/Subprogram       Line
------      ---- ----------       ----

d1cbessk    cbessl.f/d1cbessk       293D
          fluids.f/ofl_pot        79
d1cbessy    c1.f/c1a_pot            76, 84, 91
            " /c1b_pot           162, 171, 178
          c2.f/c2b_pot           80, 88, 95
            " /c2c_pot           172, 181, 188
          cbessl.f/d1cbessh1       517
            "  /d1cbessh2       532
            "  /d1cbessy        356D
d2cbessi      "  /d2cbessi        468D
d2cbessj    c1.f/c1a_pot            72, 80, 87
            " /c1b_pot           158, 167, 174
          c2.f/c2b_pot           76, 84, 91
            " /c2c_pot           168, 177, 184
          cbessl.f/d2cbessj       338D
          rf.f/rod_pot           42, 48, 52
d2cbessk    cbessl.f/d2cbessk       308D
d2cbessy    c1.f/c1a_pot            75, 83, 90
            " /c1b_pot           161, 170, 177
          c2.f/c2b_pot           79, 87, 94
            " /c2c_pot           171, 180, 187
          cbessl.f/d2cbessy       401D
dabs        c1.f/output_rc1         868, 891, 914, 940, 958, 984, 1002
          c2.f/output_rc2          1266, 1289, 1312, 1338, 1356, 1382,
                          1400
          fluids.f/ofl_pot        70, 70
          rf.f/output            358, 370, 382, 396, 405, 419, 428
dcmplx      fluids.f/ifl_pot        39
          mr2cf.f/MAIN            189, 209, 232
dsqrt       fluids.f/ofl_pot        72, 78
fac         cbessl.f/cbessi        155, 155
            "  /cbessj       88, 88
            "  /cbessk       267, 267, 279, 279
            "  /cbessy       195, 195, 207, 207
            "  /fac          21D
gamma          "  /gamma          4D
iabs          "  /cbessi        149
            "  /cbessj       71
            "  /cbessk       255
            "  /cbessy       179
            "  /psi          43
```

Subprograms :

Symbol	File/Subprogram	Line
ifl_pot	fluids.f/ifl_pot	14D
	mr2cf.f/MAIN	313, 321, 425, 440
log10	" / "	472
minv	c1.f/abc_rc1_invert	610
	c2.f/abc_rc2_invert	931
	rf.f/abc_rod_invert	237
	" /minv	448D
	smc1.f/abc_c1_invert	357
ofl_pot	fluids.f/ofl_pot	52D
	mr2cf.f/MAIN	312, 332, 354, 366, 385, 403, 430, 456
output	" / "	361, 393
	rf.f/output	264D
output_if	fluids.f/output_if	91D
	mr2cf.f/MAIN	322, 441
output_of	fluids.f/output_of	134D
	mr2cf.f/MAIN	338, 368, 406, 460
output_rc1	c1.f/output_rc1	649D
	mr2cf.f/MAIN	327, 398, 446
output_rc2	c2.f/output_rc2	991D
	mr2cf.f/MAIN	451
psi	cbessl.f/cbessk	279, 279
	" /cbessy	207, 207
	" /psi	38D
rod_pot	mr2cf.f/MAIN	353, 360, 384, 392
	rf.f/rod_pot	14D
sys_matrix_c1		
	mr2cf.f/MAIN	314
	smc1.f/sys_matrix_c1	14D
sys_matrix_rc1		
	c1.f/sys_matrix_rc1	191D
	mr2cf.f/MAIN	386
sys_matrix_rc2		
	c2.f/sys_matrix_rc2	198D
	mr2cf.f/MAIN	431
sys_matrix_rod		
	" / "	355
	rf.f/sys_matrix_rod	64D

Subprograms :

Symbol	File/Subprogram	Line
------	---- ----------	----
zabs	c1.f/output_rc1	868, 891, 914, 940, 958, 984, 1002
	c2.f/output_rc2	1266, 1289, 1312, 1338, 1356, 1382, 1400
	rf.f/minv	483, 483, 495
	" /output	358, 370, 382, 396, 405, 419, 428
zsqrt	c1.f/c1a_pot	71, 79
	" /c1b_pot	157, 166
	c2.f/c2b_pot	75, 83
	" /c2c_pot	167, 176
	fluids.f/ifl_pot	40
	mr2cf.f/MAIN	192, 193, 212, 213, 235, 236, 284
	rf.f/rod_pot	41, 47

VARIABLES

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

a        c2.f/abc_rc2_invert     800D/*, 806D
          " /sys_matrix_rc2     325D, 328=, 670, 673, 677, 680, 709,
                                711, 714, 717, 738, 738, 741, 741, 744,
                                747, 751, 754, 779, 781
         cbessl.f/cbessh1       483D/*, 485D, 489, 489
           "  /cbessh2          497D/*, 500D, 503, 503
           "  /cbessi           141D/*, 145D, 148
           "  /cbessj           58D/*, 62D, 70
           "  /cbessk           244D/*, 248D, 253, 273
           "  /cbessy           167D/*, 171D, 177, 201
           "  /d1cbessh1        511D/*, 514D, 517, 518
           "  /d1cbessh2        526D/*, 529D, 532, 533
           "  /d1cbessi         453D/*, 456D, 460, 460, 460
           "  /d1cbessj         323D/*, 326D, 330, 330, 330
           "  /d1cbessk         293D/*, 296D, 300, 300, 300
           "  /d1cbessy         356D/*, 359D, 367, 367, 371, 371, 371,
                                372, 372, 376, 376, 376, 377, 377, 378,
                                378, 379, 384, 384, 384, 388, 388, 388,
                                389, 389
           "  /d2cbessi         468D/*, 471D
           "  /d2cbessj         338D/*, 341D, 346, 346, 347, 347
           "  /d2cbessk         308D/*, 311D
           "  /d2cbessy         401D/*, 404D, 413, 413, 413, 414, 414,
                                414, 418, 418, 418, 419, 419, 420, 420,
                                421, 421, 425, 425, 425, 426, 426, 427,
                                427, 428, 428, 429, 429, 429, 430, 430,
                                435, 435, 435, 435, 436, 440, 440, 440,
                                441, 441, 441, 441
         mr2cf.f/MAIN           8D
         smc1.f/abc_c1_invert   279D/*, 285D
          " /sys_matrix_c1      87D, 90=, 178, 181, 184, 187, 205, 207,
                                225, 225, 228, 228, 231, 234, 238, 241,
                                256, 258
a1       cbessl.f/cbessj        66D, 104=, 106, 113
```

Variables :

Symbol	File/Subprogram	Line
al_c1	c1.f/abc_rc1_invert	560D, 575D, 618=, 629=
	" /c1a_pot	42D, 57D
	" /c1b_pot	128D, 143D
	" /output_rc1	675D, 690D, 751, 777, 799, 817, 836,
		855, 877, 900, 928, 944, 970, 988
	" /sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 945=, 966=
	" /output_rc2	1017D, 1032D
	" /sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 367=, 373, 381=, 387
	" /sys_matrix_c1	42D, 57D
al_c2	c2.f/abc_rc2_invert	866D, 882D, 939=, 952, 960=, 973
	" /c2b_pot	45D, 60D
	" /c2c_pot	136D, 152D
	" /output_rc2	1056D, 1072D, 1150, 1176, 1198, 1216,
		1235, 1254, 1275, 1298, 1326, 1342,
		1368, 1386
	" /sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
al_rod	c1.f/abc_rc1_invert	583D, 589D, 624=, 635=
	" /output_rc1	698D, 704D
	" /sys_matrix_rc1	242D, 248D
	c2.f/abc_rc2_invert	892D, 898D
	" /output_rc2	1081D, 1087D
	" /sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D
	rf.f/abc_rod_invert	211D, 217D, 245=, 250=
	" /abc_rod_solve	161D, 167D, 177=, 181=, 191, 191=
	" /output	274D, 280D, 312, 315, 325, 329, 336,
		342, 348, 354, 366, 377, 392, 400, 415,
		423
	" /rod_pot	25D, 30D
	" /sys_matrix_rod	74D, 80D
alc1	c1.f/output_rc1	721D, 751=, 769, 777=, 791, 799=, 809,
		817=, 829, 836=, 848, 855=, 865, 877=,
		889, 900=, 912, 928=, 938, 944=, 956,
		970=, 982, 988=, 1000

Variables :

```
Symbol     File/Subprogram        Line
------     ---- ----------        ----

a1c2       c2.f/output_rc2        1116D, 1150=, 1168, 1176=, 1190, 1198=,
                                  1208, 1216=, 1228, 1235=, 1247, 1254=,
                                  1264, 1275=, 1287, 1298=, 1310, 1326=,
                                  1336, 1342=, 1354, 1368=, 1380, 1386=,
                                  1398
a2         c1.f/sys_matrix_rc1    263D, 266=, 377, 380, 382, 385, 393,
                                  412, 415, 422, 436, 439, 447, 450
           c2.f/sys_matrix_rc2    325D, 331=, 671, 674, 676, 679, 713,
                                  714, 716, 717, 745, 748
           cbessl.f/cbessj        66D, 106=, 108, 113
           smc1.f/sys_matrix_c1   87D, 92=, 179, 182, 184, 187, 209, 212,
                                  232, 235
a2_c1      c1.f/abc_rc1_invert    560D, 575D, 619=, 630=
            " /c1a_pot            42D, 57D
            " /c1b_pot            128D, 143D
            " /output_rc1         675D, 690D, 755, 781, 801, 819, 838,
                                  857, 879, 902, 930, 946, 972, 990
            " /sys_matrix_rc1     219D, 234D
           c2.f/abc_rc2_invert    827D, 842D, 946=, 967=
            " /output_rc2         1017D, 1032D
            " /sys_matrix_rc2     225D, 241D
           mr2cf.f/MAIN           108D, 123D
           smc1.f/abc_c1_invert   306D, 321D, 368=, 374, 382=, 388
            " /sys_matrix_c1      42D, 57D
a2_c2      c2.f/abc_rc2_invert    866D, 882D, 940=, 953, 961=, 974
            " /c2b_pot            45D, 60D
            " /c2c_pot            136D, 152D
            " /output_rc2         1056D, 1072D, 1154, 1180, 1200, 1218,
                                  1237, 1256, 1277, 1300, 1328, 1344,
                                  1370, 1388
            " /sys_matrix_rc2     267D, 283D
           mr2cf.f/MAIN           149D, 164D
a2c1       c1.f/output_rc1        721D, 755=, 769, 781=, 791, 801=, 809,
                                  819=, 829, 838=, 848, 857=, 865, 879=,
                                  889, 902=, 912, 930=, 938, 946=, 956,
                                  972=, 982, 990=, 1000
a2c2       c2.f/output_rc2        1116D, 1154=, 1168, 1180=, 1190, 1200=,
                                  1208, 1218=, 1228, 1237=, 1247, 1256=,
                                  1264, 1277=, 1287, 1300=, 1310, 1328=,
                                  1336, 1344=, 1354, 1370=, 1380, 1388=,
                                  1398
```

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
a3	cbessl.f/cbessj	66D, 108=, 110, 113
a4	" / "	66D, 110=, 113
am	" / "	62D, 113=, 126
	" /cbessy	172D, 225=, 232
ao_1cyl	c1.f/sys_matrix_rc1	191D/*, 197D, 376, 379, 382, 385, 413, 416, 429, 429, 432, 432, 436, 439, 442, 445, 463, 465, 481, 483, 512, 515
	c2.f/sys_matrix_rc2	198D/*, 204D, 328, 331
	mr2cf.f/MAIN	25D, 202=, 214, 299, 310=, 313=, 314=, 316=, 318, 324, 382=, 386=, 421=, 425=, 431=, 433=, 437, 443
	smc1.f/sys_matrix_c1	14D/*, 20D, 90, 92, 210, 213
ao_2cyl	mr2cf.f/MAIN	26D, 225=, 237, 423=
ao_rod	c1.f/sys_matrix_rc1	191D/*, 197D, 266, 392, 395, 396, 408, 410, 420, 423, 447, 449, 453, 473, 493, 522
	mr2cf.f/MAIN	20D, 266=, 284, 353=, 354=, 355=, 358, 364, 384=, 386=, 390, 395
	rf.f/sys_matrix_rod	64D/*, 69D, 97, 103, 107, 113, 116, 116, 118, 121, 125, 125, 139
arg	fluids.f/ifl_pot	37D, 39=, 40
b	c1.f/abc_rc1_invert	533D/*, 539D
	c2.f/sys_matrix_rc2	325D, 329=, 451, 454, 457, 460, 467, 470, 474, 477, 493, 495, 498, 501, 507, 509, 512, 512, 515, 515, 523, 523, 526, 526, 529, 532, 536, 539, 541, 541, 544, 544, 547, 550, 554, 554, 557, 557, 569, 571, 581, 583, 595, 597, 607, 609, 634, 637, 647, 649
	smc1.f/abc_c1_invert	279D/*, 285D, 375, 376, 389, 390
	" /sys_matrix_c1	87D, 91=, 111, 115
b1	cbessl.f/cbessj	66D, 115=, 117, 124

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
b1_c1	c1.f/abc_rc1_invert	560D, 575D, 620=, 631=
	" /c1a_pot	42D, 57D
	" /c1b_pot	128D, 143D
	" /output_rc1	675D, 690D, 759, 821, 840, 881, 904, 948, 974, 992
	" /sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 947=, 968=
	" /output_rc2	1017D, 1032D
	" /sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 369=, 375, 383=, 389
	" /sys_matrix_c1	42D, 57D
b1_c2	c2.f/abc_rc2_invert	866D, 882D, 941=, 954, 962=, 975
	" /c2b_pot	45D, 60D
	" /c2c_pot	136D, 152D
	" /output_rc2	1056D, 1072D, 1158, 1220, 1239, 1279, 1302, 1346, 1372, 1390
	" /sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
b1_rod	c1.f/abc_rc1_invert	583D, 589D, 625=, 636=
	" /output_rc1	698D, 704D
	" /sys_matrix_rc1	242D, 248D
	c2.f/abc_rc2_invert	892D, 898D
	" /output_rc2	1081D, 1087D
	" /sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D
	rf.f/abc_rod_invert	211D, 217D, 246=, 251=
	" /abc_rod_solve	161D, 167D, 178=, 182=, 192, 192=
	" /output	274D, 280D, 315, 316, 342, 348, 367, 378, 401, 416, 424
	" /rod_pot	25D, 30D
	" /sys_matrix_rod	74D, 80D
b1c1	c1.f/output_rc1	721D, 759=, 769, 821=, 829, 840=, 848, 881=, 889, 904=, 912, 948=, 956, 974=, 982, 992=, 1000
b1c2	c2.f/output_rc2	1116D, 1158=, 1168, 1220=, 1228, 1239=, 1247, 1279=, 1287, 1302=, 1310, 1346=, 1354, 1372=, 1380, 1390=, 1398
b2	c1.f/sys_matrix_rc1	263D, 267=, 276, 279, 281, 284, 309, 310, 312, 313, 332, 335

Variables :

Symbol	File/Subprogram	Line
	c2.f/sys_matrix_rc2	325D, 332=, 452, 455, 457, 460, 468, 471, 473, 476, 497, 500, 511, 511, 514, 514, 530, 533, 548, 551
	cbessl.f/cbessj	66D, 117=, 119, 124
	rf.f/sys_matrix_rod	94D, 97=, 104, 106, 115, 115, 118, 122
	smc1.f/sys_matrix_c1	87D, 93=, 102, 106, 109, 113, 137, 138, 140, 141, 157, 160
b2_c1	c1.f/abc_rc1_invert	560D, 575D, 621=, 632=
	" /c1a_pot	42D, 57D
	" /c1b_pot	128D, 143D
	" /output_rc1	675D, 690D, 762, 823, 842, 883, 906, 950, 976, 994
	" /sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 948=, 969=
	" /output_rc2	1017D, 1032D
	" /sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 370=, 376, 384=, 390
	" /sys_matrix_c1	42D, 57D
b2_c2	c2.f/abc_rc2_invert	866D, 882D, 942=, 955, 963=, 976
	" /c2b_pot	45D, 60D
	" /c2c_pot	136D, 152D
	" /output_rc2	1056D, 1072D, 1161, 1222, 1241, 1281, 1304, 1348, 1374, 1392
	" /sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
b2c1	c1.f/output_rc1	721D, 762=, 769, 823=, 829, 842=, 848, 883=, 889, 906=, 912, 950=, 956, 976=, 982, 994=, 1000
b2c2	c2.f/output_rc2	1116D, 1161=, 1168, 1222=, 1228, 1241=, 1247, 1281=, 1287, 1304=, 1310, 1348=, 1354, 1374=, 1380, 1392=, 1398
b3	cbessl.f/cbessj	66D, 119=, 121, 124
b4	" / "	66D, 121=, 124
bb	mr2cf.f/MAIN	9D
bm	cbessl.f/cbessj	62D, 124=, 127
	" /cbessy	172D, 229=, 233
bmag	rf.f/minv	461D, 495=, 499

Variables :

Symbol	File/Subprogram	Line
bo_c1	c1.f/sys_matrix_rc1	191D/*, 197D, 267, 275, 278, 281, 284, 305, 307, 310, 313, 325, 325, 328, 328, 331, 334, 338, 341, 358, 360
	c2.f/sys_matrix_rc2	198D/*, 204D, 329, 332
	mr2cf.f/MAIN	47D, 214=, 299, 311=, 312=, 314=, 316=, 324, 383=, 385=, 386=, 388=, 395, 401, 422=, 431=, 443, 448
	smc1.f/sys_matrix_c1	14D/*, 20D, 91, 93, 101, 105, 109, 113, 133, 135, 138, 141, 150, 150, 153, 153, 156, 159, 163, 166
c	c2.f/abc_rc2_invert	800D/*, 806D, 954, 955, 975, 976
	" /sys_matrix_rc2	325D, 330=, 343, 347, 351, 353, 355, 357, 387, 389, 392, 395, 415, 415, 418, 418, 421, 424, 428, 431
	rf.f/minv	448D/*, 456D, 465
c1	" / "	460D, 502=, 503, 513=, 515, 521=, 523
c1_c1	c1.f/abc_rc1_invert	560D, 575D, 622=, 633=
	" /c1a_pot	42D, 57D
	" /c1b_pot	128D, 143D
	" /output_rc1	675D, 690D, 765, 785, 803, 825, 844, 859, 885, 908, 932, 952, 978, 996
	" /sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 949=, 970=
	" /output_rc2	1017D, 1032D
	" /sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 371=, 377, 385=, 391
	" /sys_matrix_c1	42D, 57D
c1_c2	c2.f/abc_rc2_invert	866D, 882D, 943=, 956, 964=, 977
	c2.f/c2b_pot	45D, 60D
	" /c2c_pot	136D, 152D
	" /output_rc2	1056D, 1072D, 1164, 1184, 1202, 1224, 1243, 1258, 1283, 1306, 1330, 1350, 1376, 1394
	" /sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
c1_rod	c1.f/abc_rc1_invert	583D, 589D, 626=, 637=
	" /output_rc1	698D, 704D
	" /sys_matrix_rc1	242D, 248D
	c2.f/abc_rc2_invert	892D, 898D
	" /output_rc2	1081D, 1087D
	" /sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D
	rf.f/abc_rod_invert	211D, 217D, 247=, 252=
	" /abc_rod_solve	161D, 167D, 179=, 183=, 193, 193=
	" /output	274D, 280D, 317, 329, 337, 343, 349, 355, 356, 368, 380, 393, 394, 403, 417, 426
	" /rod_pot	25D, 30D
	" /sys_matrix_rod	74D, 80D
c1c1	c1.f/output_rc1	721D, 765=, 769, 785=, 791, 803=, 809, 825=, 829, 844=, 848, 859=, 865, 885=, 889, 908=, 912, 932=, 938, 952=, 956, 978=, 982, 996=, 1000
c1c2	c2.f/output_rc2	1116D, 1164=, 1168, 1184=, 1190, 1202=, 1208, 1224=, 1228, 1243=, 1247, 1258=, 1264, 1283=, 1287, 1306=, 1310, 1330=, 1336, 1350=, 1354, 1376=, 1380, 1394=, 1398
c2	" /sys_matrix_rc2	325D, 333=, 344, 348, 351, 355, 391, 392, 394, 395, 422, 425
c2_c1	c1.f/abc_rc1_invert	560D, 575D, 623=, 634=
	" /c1a_pot	42D, 57D
	" /c1b_pot	128D, 143D
	" /output_rc1	675D, 690D, 767, 788, 806, 827, 846, 862, 887, 910, 935, 954, 980, 998
	" /sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 950=, 971=
	" /output_rc2	1017D, 1032D
	" /sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 372=, 378, 386=, 392
	" /sys_matrix_c1	42D, 57D

Variables :

Symbol	File/Subprogram	Line
c2_c2	c2.f/abc_rc2_invert	866D, 882D, 944=, 957, 965=, 978
	" /c2b_pot	45D, 60D
	" /c2c_pot	136D, 152D
	" /output_rc2	1056D, 1072D, 1166, 1187, 1205, 1226, 1245, 1261, 1285, 1308, 1333, 1352, 1378, 1396
	" /sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
c2c1	c1.f/output_rc1	721D, 767=, 769, 788=, 791, 806=, 809, 827=, 829, 846=, 848, 862=, 865, 887=, 889, 910=, 912, 935=, 938, 954=, 956, 980=, 982, 998=, 1000
c2c2	c2.f/output_rc2	1116D, 1166=, 1168, 1187=, 1190, 1205=, 1208, 1226=, 1228, 1245=, 1247, 1261=, 1264, 1285=, 1287, 1308=, 1310, 1333=, 1336, 1352=, 1354, 1378=, 1380, 1396=, 1398
cbessh1	cbessl.f/cbessh1	487D, 489=
	fluids.f/ofl_pot	68D
	mr2cf.f/MAIN	38D
cbessh2	cbessl.f/cbessh2	501D, 503=
	mr2cf.f/MAIN	38D
cbessi	cbessl.f/cbessi	146D, 158=
	" /cbessk	249D
	" /d1cbessi	458D
	mr2cf.f/MAIN	38D
cbessj	c1.f/c1a_pot	61D
	" /c1b_pot	147D
	c2.f/c2b_pot	64D
	" /c2c_pot	157D
	cbessl.f/cbessh1	487D
	" /cbessh2	501D
	" /cbessj	63D, 91=, 94, 94=, 126=
	" /cbessy	173D
	" /d1cbessj	328D
	" /d2cbessj	343D
	fluids.f/ifl_pot	32D
	mr2cf.f/MAIN	38D
	rf.f/rod_pot	33D

Variables :

Symbol	File/Subprogram	Line
cbessk	cbessl.f/cbessk	249D, 285=
	" /d1cbessk	298D
	fluids.f/ofl_pot	68D
	mr2cf.f/MAIN	38D
cbessy	c1.f/c1a_pot	61D
	" /c1b_pot	147D
	c2.f/c2b_pot	64D
	" /c2c_pot	157D
	cbessl.f/cbessh1	487D
	" /cbessh2	501D
	" /cbessy	173D, 213=, 217, 217=, 232=
	" /d1cbessy	361D
	" /d2cbessy	406D
	mr2cf.f/MAIN	38D
ci	c2.f/sys_matrix_rc2	199D/*, 204D
	fluids.f/ifl_pot	14D/*, 19D, 39
	mr2cf.f/MAIN	23D, 277=, 313=, 315=, 321=, 425=, 432=, 440=
	smc1.f/sys_matrix_c1	15D/*, 20D
cinv	rf.f/minv	448D/*, 456D, 530=
cl_c1	c1.f/abc_rc1_invert	559D, 574D
	" /c1a_pot	40D, 56D, 71
	" /c1b_pot	126D, 142D, 157
	" /output_rc1	674D, 689D
	" /sys_matrix_rc1	217D, 233D
	c2.f/abc_rc2_invert	826D, 841D
	" /output_rc2	1016D, 1031D
	" /sys_matrix_rc2	224D, 240D
	mr2cf.f/MAIN	107D, 122D, 212=
	smc1.f/abc_c1_invert	305D, 320D
	" /sys_matrix_c1	40D, 56D
cl_c2	c2.f/abc_rc2_invert	864D, 881D
	" /c2b_pot	43D, 59D, 75
	" /c2c_pot	134D, 151D, 167
	" /output_rc2	1054D, 1071D
	" /sys_matrix_rc2	265D, 282D
	mr2cf.f/MAIN	147D, 163D, 235=

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
cl_rod	" / "	34D, 192=, 284, 353=, 360=, 384=, 392=
	rf.f/rod_pot	14D/*, 21D, 41
co	c1.f/sys_matrix_rc1	192D/*, 197D
	c2.f/sys_matrix_rc2	199D/*, 204D
	fluids.f/ofl_pot	52D/*, 55D, 70, 72, 78
	mr2cf.f/MAIN	21D, 274=, 312=, 315=, 332=, 354=, 355=, 366=, 385=, 386=, 403=, 430=, 431=, 456=
	rf.f/sys_matrix_rod	64D/*, 69D
	smc1.f/sys_matrix_c1	15D/*, 20D
co_c2	c2.f/sys_matrix_rc2	198D/*, 204D, 330, 333
	mr2cf.f/MAIN	53D, 237=, 299, 330, 424=, 430=, 431=, 433=, 448, 454
ct_c1	c1.f/abc_rc1_invert	559D, 574D
	" /c1a_pot	40D, 56D, 79
	" /c1b_pot	126D, 142D, 166
	" /output_rc1	674D, 689D
	" /sys_matrix_rc1	217D, 233D
	c2.f/abc_rc2_invert	826D, 841D
	" /output_rc2	1016D, 1031D
	" /sys_matrix_rc2	224D, 240D
	mr2cf.f/MAIN	107D, 122D, 213=
	smc1.f/abc_c1_invert	305D, 320D
	" /sys_matrix_c1	40D, 56D
ct_c2	c2.f/abc_rc2_invert	864D, 881D
	" /c2b_pot	43D, 59D, 83
	" /c2c_pot	134D, 151D, 176
	" /output_rc2	1054D, 1071D
	" /sys_matrix_rc2	265D, 282D
	mr2cf.f/MAIN	147D, 163D, 236=
ct_rod	" / "	34D, 193=, 353=, 360=, 384=, 392=
	rf.f/rod_pot	14D/*, 21D, 47

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
cylinder1	c1.f/abc_rc1_invert	562D
	" /c1a_pot	44D
	" /c1b_pot	130D
	" /output_rc1	677D
	" /sys_matrix_rc1	221D
	c2.f/abc_rc2_invert	829D
	" /output_rc2	1019D
	" /sys_matrix_rc2	228D
	mr2cf.f/MAIN	110D
	smc1.f/abc_c1_invert	308D
	" /sys_matrix_c1	44D
cylinder2	c2.f/abc_rc2_invert	869D
	" /c2b_pot	47D
	" /c2c_pot	139D
	" /output_rc2	1059D
	" /sys_matrix_rc2	270D
	mr2cf.f/MAIN	151D
d1_ifsc	c2.f/abc_rc2_invert	915D, 917D
	" /output_rc2	1104D, 1106D
	" /sys_matrix_rc2	315D, 317D, 787
	fluids.f/ifl_pot	25D, 27D, 42=
	" /output_if	102D, 104D, 123
	mr2cf.f/MAIN	172D, 174D, 427
	smc1.f/abc_c1_invert	341D, 343D
	" /sys_matrix_c1	77D, 79D, 264
d1_ofsc	c1.f/abc_rc1_invert	595D, 597D
	" /output_rc1	710D, 712D
	" /sys_matrix_rc1	254D, 256D, 372
	c2.f/abc_rc2_invert	904D, 906D, 957, 978
	" /output_rc2	1093D, 1095D
	" /sys_matrix_rc2	305D, 307D, 345, 349, 353, 357, 360, 363
	fluids.f/ofl_pot	61D, 63D, 73=, 79=
	" /output_of	145D, 147D, 165
	mr2cf.f/MAIN	82D, 84D, 334, 367, 404, 435, 458
	rf.f/abc_rod_invert	223D, 225D
	" /sys_matrix_rod	86D, 88D, 143
	smc1.f/abc_c1_invert	330D, 332D, 378, 392
	" /sys_matrix_c1	66D, 68D, 103, 107, 111, 115, 118, 121

Variables :

```
Symbol      File/Subprogram         Line
------      ---- ----------         ----

dl_sp_cyl_a1  c1.f/abc_rc1_invert      544D, 562D
          c1.f/c1a_pot            25D, 44D, 73=
           " /c1b_pot             111D, 130D
           " /output_rc1          659D, 677D, 752, 777, 836, 877, 970
           " /sys_matrix_rc1      202D, 221D, 377, 404, 430, 459
          c2.f/abc_rc2_invert     811D, 829D
           " /output_rc2          1001D, 1019D
           " /sys_matrix_rc2      209D, 228D, 671, 705, 739, 775
          mr2cf.f/MAIN            92D, 110D
          smc1.f/abc_c1_invert    290D, 308D
           " /sys_matrix_c1       25D, 44D, 179, 201, 226, 252
dl_sp_cyl_a2  c1.f/abc_rc1_invert      545D, 563D
           " /c1a_pot             26D, 45D, 76=
           " /c1b_pot             112D, 131D
           " /output_rc1          660D, 678D, 756, 781, 838, 879, 972
           " /sys_matrix_rc1      203D, 222D, 380, 406, 433, 461
          c2.f/abc_rc2_invert     812D, 830D
           " /output_rc2          1002D, 1020D
           " /sys_matrix_rc2      210D, 229D, 674, 707, 742, 777
          mr2cf.f/MAIN            93D, 111D
          smc1.f/abc_c1_invert    291D, 309D
           " /sys_matrix_c1       26D, 45D, 182, 203, 229, 254
dl_sp_cyl_b1  c1.f/abc_rc1_invert      546D, 564D
           " /c1a_pot             27D, 46D
           " /c1b_pot             113D, 132D, 159=
           " /output_rc1          661D, 679D
           " /sys_matrix_rc1      204D, 223D, 276, 301, 326, 354
          c2.f/abc_rc2_invert     813D, 831D
           " /output_rc2          1003D, 1021D
           " /sys_matrix_rc2      211D, 230D, 468, 503, 542, 577
          mr2cf.f/MAIN            94D, 112D
          smc1.f/abc_c1_invert    292D, 310D, 373, 387
           " /sys_matrix_c1       27D, 46D, 102, 103, 129, 151
```

Variables :

```
Symbol      File/Subprogram       Line
------      ---- ----------       ----

d1_sp_cy1_b2  c1.f/abc_rc1_invert     547D, 565D
          " /c1a_pot           28D, 47D
          " /c1b_pot           114D, 133D, 162=
          " /output_rc1        662D, 680D
          " /sys_matrix_rc1    205D, 224D, 279, 303, 329, 356
        c2.f/abc_rc2_invert    814D, 832D
          " /output_rc2        1004D, 1022D
          " /sys_matrix_rc2    212D, 231D, 471, 505, 545, 579
        mr2cf.f/MAIN           95D, 113D
        smc1.f/abc_c1_invert   293D, 311D, 374, 388
          " /sys_matrix_c1     28D, 47D, 106, 107, 131, 154
d1_sp_cy2_b1  c2.f/abc_rc2_invert     849D, 869D
          " /c2b_pot           28D, 47D, 77=
          " /c2c_pot           119D, 139D
          " /output_rc2        1039D, 1059D, 1151, 1176, 1235, 1275,
                    1368
          " /sys_matrix_rc2    250D, 270D, 452, 489, 524, 565
        mr2cf.f/MAIN           132D, 151D
d1_sp_cy2_b2  c2.f/abc_rc2_invert     850D, 870D
          " /c2b_pot           29D, 48D, 80=
          " /c2c_pot           120D, 140D
          " /output_rc2        1040D, 1060D, 1155, 1180, 1237, 1277,
                    1370
          " /sys_matrix_rc2    251D, 271D, 455, 491, 527, 567
        mr2cf.f/MAIN           133D, 152D
d1_sp_cy2_c1  c2.f/abc_rc2_invert     851D, 871D, 952, 973
          " /c2b_pot           30D, 49D
          " /c2c_pot           121D, 141D, 169=
          " /output_rc2        1041D, 1061D
          " /sys_matrix_rc2    252D, 272D, 344, 345, 383, 416
        mr2cf.f/MAIN           134D, 153D
d1_sp_cy2_c2  c2.f/abc_rc2_invert     852D, 872D, 953, 974
        c2.f/c2b_pot           31D, 50D
          " /c2c_pot           122D, 142D, 172=
          " /output_rc2        1042D, 1062D
          " /sys_matrix_rc2    253D, 273D, 348, 349, 385, 419
        mr2cf.f/MAIN           135D, 154D
```

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

d1_sp_rod   c1.f/abc_rc1_invert    581D, 586D
             " /output_rc1         696D, 701D
             " /sys_matrix_rc1     240D, 245D, 393, 418, 447, 471
            c2.f/abc_rc2_invert    890D, 895D
             " /output_rc2         1079D, 1084D
             " /sys_matrix_rc2     291D, 296D
            mr2cf.f/MAIN           67D, 72D
            rf.f/abc_rod_invert    209D, 214D
             " /abc_rod_solve      159D, 164D
             " /output             272D, 277D, 312, 325, 348, 366, 415
             " /rod_pot            23D, 27D, 43=
             " /sys_matrix_rod     72D, 77D, 103, 111, 119, 137
d1_vrtp_cy1_a1
            c1.f/abc_rc1_invert    554D, 570D
             " /c1a_pot            35D, 52D, 88=
             " /c1b_pot            121D, 138D
             " /output_rc1         669D, 685D, 765, 785, 803, 859, 908,
                                   932, 952, 996
             " /sys_matrix_rc1     212D, 229D, 388, 413, 441, 511
            c2.f/abc_rc2_invert    821D, 837D
             " /output_rc2         1011D, 1027D
             " /sys_matrix_rc2     219D, 236D, 682, 714, 750
            mr2cf.f/MAIN           102D, 118D
            smc1.f/abc_c1_invert   300D, 316D
             " /sys_matrix_c1      35D, 52D, 190, 210, 237
d1_vrtp_cy1_a2
            c1.f/abc_rc1_invert    555D, 571D
             " /c1a_pot            36D, 53D, 91=
             " /c1b_pot            122D, 139D
             " /output_rc1         670D, 686D, 767, 788, 806, 862, 910,
                                   935, 954, 998
             " /sys_matrix_rc1     213D, 230D, 390, 416, 444, 514
            c2.f/abc_rc2_invert    822D, 838D
             " /output_rc2         1012D, 1028D
             " /sys_matrix_rc2     220D, 237D, 684, 717, 753
            mr2cf.f/MAIN           103D, 119D
            smc1.f/abc_c1_invert   301D, 317D
             " /sys_matrix_c1      36D, 53D, 192, 213, 240
d1_vrtp_cy1_b1
            c1.f/abc_rc1_invert    556D, 572D
             " /c1a_pot            37D, 54D
```

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

            " /c1b_pot             123D, 140D, 175=
            " /output_rc1          671D, 687D
            " /sys_matrix_rc1      214D, 231D, 287, 310, 337
          c2.f/abc_rc2_invert      823D, 839D
            " /output_rc2          1013D, 1029D
            " /sys_matrix_rc2      221D, 238D, 479, 512, 553, 647
          mr2cf.f/MAIN             104D, 120D
          smc1.f/abc_c1_invert     302D, 318D
            " /sys_matrix_c1       37D, 54D, 117, 138, 162
d1_vrtp_cy1_b2
          c1.f/abc_rc1_invert      557D, 573D
            " /c1a_pot             38D, 55D
            " /c1b_pot             124D, 141D, 178=
            " /output_rc1          672D, 688D
            " /sys_matrix_rc1      215D, 232D, 289, 313, 340
          c2.f/abc_rc2_invert      824D, 840D
            " /output_rc2          1014D, 1030D
            " /sys_matrix_rc2      222D, 239D, 481, 515, 556, 649
          mr2cf.f/MAIN             105D, 121D
          smc1.f/abc_c1_invert     303D, 319D
            " /sys_matrix_c1       38D, 55D, 120, 141, 165
d1_vrtp_cy2_b1
          c2.f/abc_rc2_invert      859D, 877D
            " /c2b_pot             38D, 55D, 92=
            " /c2c_pot             129D, 147D
            " /output_rc2          1049D, 1067D, 1164, 1184, 1202, 1258,
                                   1306, 1330, 1350, 1394
            " /sys_matrix_rc2      260D, 278D, 463, 498, 535, 633
          mr2cf.f/MAIN             142D, 159D
d1_vrtp_cy2_b2
          c2.f/abc_rc2_invert      860D, 878D
            " /c2b_pot             39D, 56D, 95=
            " /c2c_pot             130D, 148D
            " /output_rc2          1050D, 1068D, 1166, 1187, 1205, 1261,
                                   1308, 1333, 1352, 1396
            " /sys_matrix_rc2      261D, 279D, 465, 501, 538, 636
          mr2cf.f/MAIN             143D, 160D
```

Variables :

```
Symbol     File/Subprogram      Line
------     ---- ----------      ----

d1_vrtp_cy2_c1
           c2.f/abc_rc2_invert     861D, 879D
            " /c2b_pot             40D, 57D
            " /c2c_pot             131D, 149D, 185=
            " /output_rc2          1051D, 1069D
            " /sys_matrix_rc2      262D, 280D, 359, 392, 427
           mr2cf.f/MAIN            144D, 161D
d1_vrtp_cy2_c2
           c2.f/abc_rc2_invert     862D, 880D
            " /c2b_pot             41D, 58D
            " /c2c_pot             132D, 150D, 188=
            " /output_rc2          1052D, 1070D
            " /sys_matrix_rc2      263D, 281D, 362, 395, 430
           mr2cf.f/MAIN            145D, 162D
d1_vrtp_rod  c1.f/abc_rc1_invert   582D, 588D
            " /output_rc1          697D, 703D
            " /sys_matrix_rc1      241D, 247D, 398, 423, 452, 521
           c2.f/abc_rc2_invert     891D, 897D
            " /output_rc2          1080D, 1086D
            " /sys_matrix_rc2      292D, 298D
           mr2cf.f/MAIN            69D, 73D
           rf.f/abc_rod_invert     210D, 216D
            " /abc_rod_solve       160D, 166D
            " /output              273D, 279D, 317, 330, 337, 355, 380,
                                 393, 403, 426
            " /rod_pot             24D, 29D, 53=
            " /sys_matrix_rod      73D, 79D, 109, 116, 124
d1_vxp_cy1_a1
           c1.f/abc_rc1_invert     549D, 566D
            " /c1a_pot             30D, 48D, 81=
            " /c1b_pot             116D, 134D
            " /output_rc1          664D, 681D, 760, 821, 881, 904, 948,
                                 974, 992
            " /sys_matrix_rc1      207D, 225D, 383, 435, 485
           c2.f/abc_rc2_invert     816D, 833D
            " /output_rc2          1006D, 1023D
            " /sys_matrix_rc2      214D, 232D, 677, 745
           mr2cf.f/MAIN            97D, 114D
           smc1.f/abc_c1_invert    295D, 312D
            " /sys_matrix_c1       30D, 48D, 185, 232
```

Variables :

```
Symbol     File/Subprogram        Line
------     ---- ----------        ----

d1_vxp_cy1_a2
        c1.f/abc_rc1_invert     550D, 567D
         " /c1a_pot             31D, 49D, 84=
         " /c1b_pot             117D, 135D
         " /output_rc1          665D, 682D, 763, 823, 883, 906, 950,
                             976, 994
         " /sys_matrix_rc1      208D, 226D, 386, 438, 487
        c2.f/abc_rc2_invert     817D, 834D
         " /output_rc2          1007D, 1024D
         " /sys_matrix_rc2      215D, 233D, 680, 748
        mr2cf.f/MAIN            98D, 115D
        smc1.f/abc_c1_invert    296D, 313D
         " /sys_matrix_c1       31D, 49D, 188, 235
d1_vxp_cy1_b1
        c1.f/abc_rc1_invert     551D, 568D
         " /c1a_pot             32D, 50D
         " /c1b_pot             118D, 136D, 168=
         " /output_rc1          666D, 683D
         " /sys_matrix_rc1      209D, 227D, 282, 332
        c2.f/abc_rc2_invert     818D, 835D
         " /output_rc2          1008D, 1025D
         " /sys_matrix_rc2      216D, 234D, 474, 547, 611
        mr2cf.f/MAIN            99D, 116D
        smc1.f/abc_c1_invert    297D, 314D
         " /sys_matrix_c1       32D, 50D, 110, 157
d1_vxp_cy1_b2
        c1.f/abc_rc1_invert     552D, 569D
         " /c1a_pot             33D, 51D
         " /c1b_pot             119D, 137D, 171=
         " /output_rc1          667D, 684D
         " /sys_matrix_rc1      210D, 228D, 285, 335
        c2.f/abc_rc2_invert     819D, 836D
         " /output_rc2          1009D, 1026D
         " /sys_matrix_rc2      217D, 235D, 477, 550, 613
        mr2cf.f/MAIN            100D, 117D
        smc1.f/abc_c1_invert    298D, 315D
         " /sys_matrix_c1       33D, 51D, 114, 160
```

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

d1_vxp_cy2_b1
            c2.f/abc_rc2_invert    854D, 873D
             " /c2b_pot            33D, 51D, 85=
             " /c2c_pot            124D, 143D
             " /output_rc2         1044D, 1063D, 1159, 1220, 1279, 1302,
                                 1346, 1372, 1390
             " /sys_matrix_rc2     255D, 274D, 458, 529, 599
            mr2cf.f/MAIN           137D, 155D
d1_vxp_cy2_b2
            c2.f/abc_rc2_invert    855D, 874D
             " /c2b_pot            34D, 52D, 88=
             " /c2c_pot            125D, 144D
             " /output_rc2         1045D, 1064D, 1162, 1222, 1281, 1304,
                                 1348, 1374, 1392
             " /sys_matrix_rc2     256D, 275D, 461, 532, 601
            mr2cf.f/MAIN           138D, 156D
d1_vxp_cy2_c1
            c2.f/abc_rc2_invert    856D, 875D
             " /c2b_pot            35D, 53D
             " /c2c_pot            126D, 145D, 178=
             " /output_rc2         1046D, 1065D
             " /sys_matrix_rc2     257D, 276D, 352, 422
            mr2cf.f/MAIN           139D, 157D
d1_vxp_cy2_c2
            c2.f/abc_rc2_invert    857D, 876D
             " /c2b_pot            36D, 54D
             " /c2c_pot            127D, 146D, 181=
             " /output_rc2         1047D, 1066D
             " /sys_matrix_rc2     258D, 277D, 356, 425
            mr2cf.f/MAIN           140D, 158D
```

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
d1_vxp_rod	c1.f/abc_rc1_invert	582D, 587D
	“ /output_rc1	697D, 702D
	c1.f/sys_matrix_rc1	241D, 246D, 395, 449, 495
	c2.f/abc_rc2_invert	891D, 896D
	“ /output_rc2	1080D, 1085D
	“ /sys_matrix_rc2	292D, 297D
	mr2cf.f/MAIN	68D, 73D
	rf.f/abc_rod_invert	210D, 215D
	“ /abc_rod_solve	160D, 165D
	“ /output	273D, 278D, 316, 342, 367, 379, 402, 416, 425
	“ /rod_pot	24D, 28D, 49=
	“ /sys_matrix_rod	73D, 78D, 107, 122
d1cbessh1	cbessl.f/d1cbessh1	515D, 517=
	fluids.f/ofl_pot	68D
	mr2cf.f/MAIN	43D
d1cbessh2	cbessl.f/d1cbessh2	530D, 532=
	mr2cf.f/MAIN	43D
d1cbessi	cbessl.f/d1cbessi	458D, 460=
	mr2cf.f/MAIN	42D
d1cbessj	c1.f/c1a_pot	62D
	“ /c1b_pot	148D
	c2.f/c2b_pot	65D
	“ /c2c_pot	158D
	cbessl.f/d1cbessh1	515D
	“ /d1cbessh2	530D
	“ /d1cbessj	328D, 330=
	fluids.f/ifl_pot	32D
	mr2cf.f/MAIN	39D
	rf.f/rod_pot	34D
d1cbessk	cbessl.f/d1cbessk	298D, 300=
	fluids.f/ofl_pot	68D
	mr2cf.f/MAIN	41D
d1cbessy	c1.f/c1a_pot	63D
	“ /c1b_pot	149D
	c2.f/c2b_pot	66D
	“ /c2c_pot	159D
	cbessl.f/d1cbessh1	515D
	“ /d1cbessh2	530D
	“ /d1cbessy	361D, 367=, 372=, 379=, 384=, 389=
	mr2cf.f/MAIN	40D

Variables :

```
Symbol       File/Subprogram        Line
------       ---- ----------        ----

d2_sp_cyl_a1  c1.f/abc_rc1_invert     544D, 562D
              " /c1a_pot              25D, 44D, 72=
              " /c1b_pot              111D, 130D
              " /output_rc1           659D, 677D, 751, 777, 900, 944, 988
              " /sys_matrix_rc1       202D, 221D, 376
             c2.f/abc_rc2_invert      811D, 829D
              " /output_rc2           1001D, 1019D
              " /sys_matrix_rc2       209D, 228D, 670
             mr2cf.f/MAIN             92D, 110D
             smc1.f/abc_c1_invert     290D, 308D
              "  /sys_matrix_c1       25D, 44D, 178
d2_sp_cyl_a2  c1.f/abc_rc1_invert     545D, 563D
              " /c1a_pot              26D, 45D, 75=
              " /c1b_pot              112D, 131D
              " /output_rc1           660D, 678D, 755, 781, 902, 946, 990
              " /sys_matrix_rc1       203D, 222D, 379
             c2.f/abc_rc2_invert      812D, 830D
              " /output_rc2           1002D, 1020D
              " /sys_matrix_rc2       210D, 229D, 673
             mr2cf.f/MAIN             93D, 111D
             smc1.f/abc_c1_invert     291D, 309D
              "  /sys_matrix_c1       26D, 45D, 181
d2_sp_cyl_b1  c1.f/abc_rc1_invert     546D, 564D
              " /c1a_pot              27D, 46D
              " /c1b_pot              113D, 132D, 158=
             c1.f/output_rc1          661D, 679D
              " /sys_matrix_rc1       204D, 223D, 275
             c2.f/abc_rc2_invert      813D, 831D
              " /output_rc2           1003D, 1021D
              " /sys_matrix_rc2       211D, 230D, 467
             mr2cf.f/MAIN             94D, 112D
             smc1.f/abc_c1_invert     292D, 310D
              "  /sys_matrix_c1       27D, 46D, 101
```

Variables :

```
Symbol       File/Subprogram        Line
------       ---- ----------        ----

d2_sp_cy1_b2 c1.f/abc_rc1_invert    547D, 565D
              " /c1a_pot            28D, 47D
              " /c1b_pot            114D, 133D, 161=
              " /output_rc1         662D, 680D
              " /sys_matrix_rc1     205D, 224D, 278
             c2.f/abc_rc2_invert    814D, 832D
              " /output_rc2         1004D, 1022D
              " /sys_matrix_rc2     212D, 231D, 470
             mr2cf.f/MAIN           95D, 113D
             smc1.f/abc_c1_invert   293D, 311D
              " /sys_matrix_c1      28D, 47D, 105
d2_sp_cy2_b1 c2.f/abc_rc2_invert    849D, 869D
              " /c2b_pot            28D, 47D, 76=
              " /c2c_pot            119D, 139D
              " /output_rc2         1039D, 1059D, 1150, 1176, 1298, 1342,
                                    1386
              " /sys_matrix_rc2     250D, 270D, 451
             mr2cf.f/MAIN           132D, 151D
d2_sp_cy2_b2 c2.f/abc_rc2_invert    850D, 870D
              " /c2b_pot            29D, 48D, 79=
              " /c2c_pot            120D, 140D
              " /output_rc2         1040D, 1060D, 1154, 1180, 1300, 1344,
                                    1388
              " /sys_matrix_rc2     251D, 271D, 454
             mr2cf.f/MAIN           133D, 152D
d2_sp_cy2_c1 c2.f/abc_rc2_invert    851D, 871D
              " /c2b_pot            30D, 49D
              " /c2c_pot            121D, 141D, 168=
              " /output_rc2         1041D, 1061D
              " /sys_matrix_rc2     252D, 272D, 343
             mr2cf.f/MAIN           134D, 153D
d2_sp_cy2_c2 c2.f/abc_rc2_invert    852D, 872D
              " /c2b_pot            31D, 50D
              " /c2c_pot            122D, 142D, 171=
              " /output_rc2         1042D, 1062D
              " /sys_matrix_rc2     253D, 273D, 347
             mr2cf.f/MAIN           135D, 154D
```

Variables :

Symbol	File/Subprogram	Line
d2_sp_rod	c1.f/abc_rc1_invert	581D, 586D
	" /output_rc1	696D, 701D
	" /sys_matrix_rc1	240D, 245D, 392
	c2.f/abc_rc2_invert	890D, 895D
	" /output_rc2	1079D, 1084D
	" /sys_matrix_rc2	291D, 296D
	mr2cf.f/MAIN	67D, 72D
	rf.f/abc_rod_invert	209D, 214D
	" /abc_rod_solve	159D, 164D
	" /output	272D, 277D, 312, 315, 325, 377, 400, 423
	" /rod_pot	23D, 27D, 42=
	" /sys_matrix_rod	72D, 77D, 102
d2_vrtp_cy1_a1		
	c1.f/abc_rc1_invert	554D, 570D
	" /c1a_pot	35D, 52D, 87=
	" /c1b_pot	121D, 138D
	" /output_rc1	669D, 685D
	" /sys_matrix_rc1	212D, 229D, 413
	c2.f/abc_rc2_invert	821D, 837D
	" /output_rc2	1011D, 1027D
	" /sys_matrix_rc2	219D, 236D, 714
	mr2cf.f/MAIN	102D, 118D
	smc1.f/abc_c1_invert	300D, 316D
	" /sys_matrix_c1	35D, 52D, 210
d2_vrtp_cy1_a2		
	c1.f/abc_rc1_invert	555D, 571D
	" /c1a_pot	36D, 53D, 90=
	" /c1b_pot	122D, 139D
	" /output_rc1	670D, 686D
	" /sys_matrix_rc1	213D, 230D, 416
	c2.f/abc_rc2_invert	822D, 838D
	" /output_rc2	1012D, 1028D
	" /sys_matrix_rc2	220D, 237D, 717
	mr2cf.f/MAIN	103D, 119D
	smc1.f/abc_c1_invert	301D, 317D
	" /sys_matrix_c1	36D, 53D, 213

Variables :

Symbol	File/Subprogram	Line
d2_vrtp_cy1_b1		
	c1.f/abc_rc1_invert	556D, 572D
	" /c1a_pot	37D, 54D
	" /c1b_pot	123D, 140D, 174=
	" /output_rc1	671D, 687D
	" /sys_matrix_rc1	214D, 231D, 310
	c2.f/abc_rc2_invert	823D, 839D
	" /output_rc2	1013D, 1029D
	" /sys_matrix_rc2	221D, 238D, 512
	mr2cf.f/MAIN	104D, 120D
	smc1.f/abc_c1_invert	302D, 318D
	" /sys_matrix_c1	37D, 54D, 138
d2_vrtp_cy1_b2		
	c1.f/abc_rc1_invert	557D, 573D
	" /c1a_pot	38D, 55D
	" /c1b_pot	124D, 141D, 177=
	" /output_rc1	672D, 688D
	" /sys_matrix_rc1	215D, 232D, 313
	c2.f/abc_rc2_invert	824D, 840D
	" /output_rc2	1014D, 1030D
	" /sys_matrix_rc2	222D, 239D, 515
	mr2cf.f/MAIN	105D, 121D
	smc1.f/abc_c1_invert	303D, 319D
	" /sys_matrix_c1	38D, 55D, 141
d2_vrtp_cy2_b1		
	c2.f/abc_rc2_invert	859D, 877D
	" /c2b_pot	38D, 55D, 91=
	" /c2c_pot	129D, 147D
	" /output_rc2	1049D, 1067D
	" /sys_matrix_rc2	260D, 278D, 498
	mr2cf.f/MAIN	142D, 159D
d2_vrtp_cy2_b2		
	c2.f/abc_rc2_invert	860D, 878D
	" /c2b_pot	39D, 56D, 94=
	" /c2c_pot	130D, 148D
	" /output_rc2	1050D, 1068D
	" /sys_matrix_rc2	261D, 279D, 501
	mr2cf.f/MAIN	143D, 160D

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

d2_vrtp_cy2_c1
            c2.f/abc_rc2_invert     861D, 879D
             " /c2b_pot             40D, 57D
             " /c2c_pot             131D, 149D, 184=
             " /output_rc2          1051D, 1069D
             " /sys_matrix_rc2      262D, 280D, 392
            mr2cf.f/MAIN            144D, 161D
d2_vrtp_cy2_c2
            c2.f/abc_rc2_invert     862D, 880D
             " /c2b_pot             41D, 58D
             " /c2c_pot             132D, 150D, 187=
             " /output_rc2          1052D, 1070D
             " /sys_matrix_rc2      263D, 281D, 395
            mr2cf.f/MAIN            145D, 162D
d2_vrtp_rod  c1.f/abc_rc1_invert    582D, 588D
             " /output_rc1          697D, 703D
             " /sys_matrix_rc1      241D, 247D, 423
            c2.f/abc_rc2_invert     891D, 897D
             " /output_rc2          1080D, 1086D
             " /sys_matrix_rc2      292D, 298D
            mr2cf.f/MAIN            69D, 73D
            rf.f/abc_rod_invert     210D, 216D
             " /abc_rod_solve       160D, 166D
             " /output              273D, 279D
             " /rod_pot             24D, 29D, 52=
             " /sys_matrix_rod      73D, 79D, 116
d2_vxp_cy1_a1
            c1.f/abc_rc1_invert     549D, 566D
             " /c1a_pot             30D, 48D, 80=
             " /c1b_pot             116D, 134D
             " /output_rc1          664D, 681D
             " /sys_matrix_rc1      207D, 225D, 435
            c2.f/abc_rc2_invert     816D, 833D
             " /output_rc2          1006D, 1023D
             " /sys_matrix_rc2      214D, 232D, 744
            mr2cf.f/MAIN            97D, 114D
            smc1.f/abc_c1_invert    295D, 312D
              " /sys_matrix_c1      30D, 48D, 231
```

Variables :

```
Symbol     File/Subprogram      Line
------     ---- ----------      ----

d2_vxp_cy1_a2
           c1.f/abc_rc1_invert      550D, 567D
            " /c1a_pot              31D, 49D, 83=
            " /c1b_pot              117D, 135D
            " /output_rc1           665D, 682D
            " /sys_matrix_rc1       208D, 226D, 438
           c2.f/abc_rc2_invert      817D, 834D
            " /output_rc2           1007D, 1024D
            " /sys_matrix_rc2       215D, 233D, 747
           mr2cf.f/MAIN             98D, 115D
           smc1.f/abc_c1_invert     296D, 313D
            " /sys_matrix_c1        31D, 49D, 234
d2_vxp_cy1_b1
           c1.f/abc_rc1_invert      551D, 568D
            " /c1a_pot              32D, 50D
            " /c1b_pot              118D, 136D, 167=
            " /output_rc1           666D, 683D
            " /sys_matrix_rc1       209D, 227D, 331
           c2.f/abc_rc2_invert      818D, 835D
            " /output_rc2           1008D, 1025D
            " /sys_matrix_rc2       216D, 234D, 547
           mr2cf.f/MAIN             99D, 116D
           smc1.f/abc_c1_invert     297D, 314D
            " /sys_matrix_c1        32D, 50D, 156
d2_vxp_cy1_b2
           c1.f/abc_rc1_invert      552D, 569D
            " /c1a_pot              33D, 51D
            " /c1b_pot              119D, 137D, 170=
            " /output_rc1           667D, 684D
            " /sys_matrix_rc1       210D, 228D, 334
           c2.f/abc_rc2_invert      819D, 836D
            " /output_rc2           1009D, 1026D
            " /sys_matrix_rc2       217D, 235D, 550
           mr2cf.f/MAIN             100D, 117D
           smc1.f/abc_c1_invert     298D, 315D
            " /sys_matrix_c1        33D, 51D, 159
```

Variables :

```
Symbol      File/Subprogram      Line
------      ---- ----------      ----

d2_vxp_cy2_b1
            c2.f/abc_rc2_invert       854D, 873D
              " /c2b_pot              33D, 51D, 84=
              " /c2c_pot              124D, 143D
              " /output_rc2           1044D, 1063D
              " /sys_matrix_rc2       255D, 274D, 529
            mr2cf.f/MAIN              137D, 155D
d2_vxp_cy2_b2
            c2.f/abc_rc2_invert       855D, 874D
              " /c2b_pot              34D, 52D, 87=
              " /c2c_pot              125D, 144D
              " /output_rc2           1045D, 1064D
              " /sys_matrix_rc2       256D, 275D, 532
            mr2cf.f/MAIN              138D, 156D
d2_vxp_cy2_c1
            c2.f/abc_rc2_invert       856D, 875D
              " /c2b_pot              35D, 53D
              " /c2c_pot              126D, 145D, 177=
              " /output_rc2           1046D, 1065D
              " /sys_matrix_rc2       257D, 276D, 421
            mr2cf.f/MAIN              139D, 157D
d2_vxp_cy2_c2
            c2.f/abc_rc2_invert       857D, 876D
              " /c2b_pot              36D, 54D
              " /c2c_pot              127D, 146D, 180=
              " /output_rc2           1047D, 1066D
              " /sys_matrix_rc2       258D, 277D, 424
            mr2cf.f/MAIN              140D, 158D
d2_vxp_rod  c1.f/abc_rc1_invert       582D, 587D
              " /output_rc1           697D, 702D
              " /sys_matrix_rc1       241D, 246D, 449
            c2.f/abc_rc2_invert       891D, 896D
              " /output_rc2           1080D, 1085D
              " /sys_matrix_rc2       292D, 297D
            mr2cf.f/MAIN              68D, 73D
            rf.f/abc_rod_invert       210D, 215D
              " /abc_rod_solve        160D, 165D
              " /output               273D, 278D
              " /rod_pot              24D, 28D, 48=
              " /sys_matrix_rod       73D, 78D, 121
```

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
d2cbessi	cbessl.f/d2cbessi	473D, 475=
	mr2cf.f/MAIN	42D
d2cbessj	c1.f/c1a_pot	62D
	" /c1b_pot	148D
	c2.f/c2b_pot	65D
	" /c2c_pot	158D
	cbessl.f/d2cbessj	343D, 346=
	mr2cf.f/MAIN	39D
	rf.f/rod_pot	34D
d2cbessk	cbessl.f/d2cbessk	313D, 315=
	mr2cf.f/MAIN	41D
d2cbessy	c1.f/c1a_pot	63D
	" /c1b_pot	149D
	c2.f/c2b_pot	66D
	" /c2c_pot	159D
	cbessl.f/d2cbessy	406D, 414=, 421=, 430=, 435=, 441=
	mr2cf.f/MAIN	40D
d_if	c2.f/abc_rc2_invert	915D, 917D, 951=, 972=
	" /output_rc2	1104D, 1106D
	" /sys_matrix_rc2	315D, 317D
	fluids.f/ifl_pot	25D, 27D
	" /output_if	102D, 104D, 116, 123
	mr2cf.f/MAIN	172D, 174D, 427
	smc1.f/abc_c1_invert	341D, 343D, 379=, 393=
	" /sys_matrix_c1	77D, 79D
deta	rf.f/abc_rod_solve	171D, 186=, 191, 192, 193
dill	" /output	286D, 312=, 314, 325=, 328
dpp	c1.f/output_rc1	720D, 962=, 964, 1007=, 1008
	c2.f/output_rc2	1115D, 1360=, 1362, 1405=, 1406
	rf.f/output	286D, 409=, 411, 433=, 434
e_1cyl	mr2cf.f/MAIN	25D, 199=, 209
e_2cyl	" / "	26D, 222=, 232
e_rod	" / "	22D, 182=, 189
ec_c1	" / "	49D, 209=, 210, 211
ec_c2	" / "	55D, 232=, 233, 234
ec_rod	" / "	37D, 189=, 190, 191
emt	" / "	19D, 271=, 294, 307, 351, 380, 419

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
err	c1.f/output_rc1	720D, 912=, 914, 918, 956=, 958, 959=, 962, 1000=, 1002, 1003=, 1006, 1007
	c2.f/output_rc2	1115D, 1310=, 1312, 1316, 1354=, 1356, 1357=, 1360, 1398=, 1400, 1401=, 1404, 1405
	rf.f/output	286D, 377=, 382, 386, 400=, 405, 406=, 409, 423=, 428, 429=, 432, 433
ett	c1.f/output_rc1	720D, 889=, 891, 895, 982=, 984, 1006
	c2.f/output_rc2	1115D, 1287=, 1289, 1293, 1380=, 1382, 1404
	rf.f/output	286D, 366=, 370, 373, 415=, 419, 432
euler	cbessl.f/psi	40D, 41D, 45, 50
exctype	c1.f/abc_rc1_invert	533D/*, 538D, 617, 628
	c2.f/abc_rc2_invert	800D/*, 805D, 938, 959
	mr2cf.f/MAIN	19D, 270=, 316=, 356=, 388=, 433=
	rf.f/abc_rod_invert	201D/*, 206D, 244, 249
	" /abc_rod_solve	152D/*, 156D, 176, 180
	smc1.f/abc_c1_invert	279D/*, 284D, 366, 380
exx	c1.f/output_rc1	720D, 865=, 868, 872, 938=, 940, 941=, 962, 962, 1006=, 1007, 1007
	c2.f/output_rc2	1115D, 1264=, 1266, 1270, 1336=, 1338, 1339=, 1360, 1360, 1404=, 1405, 1405
	rf.f/output	286D, 354=, 358, 362, 392=, 396, 397=, 409, 409, 432=, 433, 433
ez	cbessl.f/cbessj	66D, 76=, 115
ez2	" / "	66D, 77=, 104, 106, 108, 110, 117, 119, 121
f	mr2cf.f/MAIN	10D, 294, 297
f2	fluids.f/ofl_pot	67D, 78=, 79, 80
fa	mr2cf.f/MAIN	10D
fac	cbessl.f/cbessi	143D
	" /cbessj	60D
	" /cbessk	246D
	" /cbessy	169D
	" /fac	22D, 24=, 30=
	mr2cf.f/MAIN	32D
fn2	cbessl.f/cbessj	65D, 75=, 104, 104, 106, 106, 108, 108, 110, 110, 115, 117, 117, 119, 119, 121, 121

Variables :

Symbol	File/Subprogram	Line
g	mr2cf.f/MAIN	22D, 280=, 346=, 376=, 414=, 468=, 472=, 478
g1	fluids.f/ifl_pot	37D, 40=, 42, 43
g2	" /ofl_pot	67D, 72=, 73, 74
gamma	cbessl.f/gamma	5D, 7=, 13=
	mr2cf.f/MAIN	32D
h_1cyl	" / "	25D, 204=, 214
h_2cyl	" / "	26D, 227=, 237
i	cbessl.f/fac	22D, 27=, 28
	" /gamma	5D, 10=, 11
	cbessl.f/psi	39D, 47=, 48
	rf.f/minv	459D, 463=, 465, 465, 469=, 471, 472, 472, 477=, 478, 479, 481, 491, 492, 500, 502, 502, 507, 507=, 510, 512=, 513, 513, 514, 515, 515, 519=, 520, 521, 523, 528=, 530, 530
iflag	c1.f/abc_rc1_invert	603D, 607=, 610=
	c2.f/abc_rc2_invert	924D, 928=, 931=
	mr2cf.f/MAIN	32D, 282=
	rf.f/abc_rod_invert	231D, 235=, 237=
	" /minv	448D/*, 455D, 533=, 552=
	smc1.f/abc_c1_invert	350D, 354=, 357=
ifluid	c2.f/abc_rc2_invert	915D
	" /output_rc2	1104D
	" /sys_matrix_rc2	315D
	fluids.f/ifl_pot	27D
	" /output_if	104D
	mr2cf.f/MAIN	174D
	smc1.f/abc_c1_invert	341D
	" /sys_matrix_c1	77D
ifsc	c2.f/abc_rc2_invert	915D, 917D
	" /output_rc2	1104D, 1106D
	" /sys_matrix_rc2	315D, 317D, 686
	fluids.f/ifl_pot	25D, 27D, 43=
	" /output_if	102D, 104D, 116
	mr2cf.f/MAIN	172D, 174D, 427
	smc1.f/abc_c1_invert	341D, 343D
	" /sys_matrix_c1	77D, 79D, 194
in	mr2cf.f/MAIN	8D
ip	rf.f/minv	459D, 478=, 484=, 488, 490, 491

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
iptmax	mr2cf.f/MAIN	17D, 18D, 22
irow	rf.f/minv	459D, 481=, 482, 484, 485, 500=, 502, 503, 503, 505
j	cbessl.f/cbessi	143D, 154=, 155
	" /cbessj	60D, 87=, 88
	" /cbessk	246D, 266=, 267, 276=, 279
	" /cbessy	169D, 194=, 195, 204=, 207
	rf.f/minv	459D, 464=, 465, 465, 470=, 471, 472, 472, 476=, 479, 482, 485, 501, 502, 502, 505, 508, 508=, 510, 514=, 515, 515, 520=, 521, 522, 523, 523, 529=, 530, 530
jcol	" / "	459D, 522=, 523, 523, 523
jk	mr2cf.f/MAIN	19D, 272=, 280, 346, 376, 414, 468, 472, 478
jn	" / "	8D
jrow	rf.f/minv	459D, 489=, 490, 491, 491, 492, 501=, 502, 503, 503
k	c1.f/c1a_pot	14D/*, 20D, 71, 79
	" /c1b_pot	100D/*, 106D, 157, 166
	" /output_rc1	649D/*, 654D, 725, 765, 767, 785, 788, 799, 801, 825, 827, 844, 846, 859, 862, 885, 887, 908, 910, 932, 935, 952, 954, 978, 980, 996, 998
	" /sys_matrix_rc1	191D/*, 197D, 269, 287, 289, 301, 303, 305, 307, 337, 340, 362, 364, 388, 390, 398, 404, 406, 408, 410, 418, 420, 441, 444, 452, 467, 469, 475, 489, 491, 497, 503, 505, 517
	c2.f/abc_rc2_invert	956, 957, 977, 978
	" /c2b_pot	17D/*, 23D, 75, 83
	" /c2c_pot	107D/*, 113D, 167, 176
	" /output_rc2	991D/*, 996D, 1121, 1164, 1166, 1184, 1187, 1198, 1200, 1224, 1226, 1243, 1245, 1258, 1261, 1283, 1285, 1306, 1308, 1330, 1333, 1350, 1352, 1376, 1378, 1394, 1396

Variables :

```
Symbol     File/Subprogram       Line
------     ---- ----------       ----

           " /sys_matrix_rc2    198D/*, 204D, 335, 359, 360, 362, 363,
                                383, 385, 387, 389, 427, 430, 463, 465,
                                479, 481, 489, 491, 493, 495, 503, 505,
                                507, 509, 535, 538, 553, 556, 573, 575,
                                585, 587, 603, 605, 615, 617, 625, 627,
                                639, 641, 682, 684, 705, 707, 709, 711,
                                750, 753, 783, 785
          cbessl.f/cbessi       143D, 153=, 154, 155, 155
            "  /cbessj          60D, 86=, 87, 88, 88
            "  /cbessk          246D, 265=, 266, 267, 267, 275=, 276,
                                277, 278, 279, 279
            "  /cbessy          169D, 193=, 194, 195, 195, 203=, 204,
                                205, 206, 207, 207
          fluids.f/ifl_pot      14D/*, 19D, 39
            "  /ofl_pot         52D/*, 55D, 70, 72, 78
            "  /output_if       91D/*, 96D
            "  /output_of       134D/*, 139D
          mr2cf.f/MAIN          21D, 264=, 284, 294, 310=, 311=, 312=,
                                313=, 314=, 321=, 322=, 326=, 327=,
                                332=, 338=, 353=, 354=, 355=, 360=,
                                361=, 366=, 368=, 382=, 383=, 384=,
                                385=, 386=, 392=, 393=, 397=, 398=,
                                403=, 406=, 421=, 422=, 423=, 424=,
                                425=, 430=, 431=, 440=, 441=, 445=,
                                446=, 450=, 451=, 456=, 460=, 480
          rf.f/minv             459D, 536=
           " /output            264D/*, 269D, 290, 317, 329, 336, 343,
                                349, 355, 356, 368, 380, 393, 394, 403,
                                417, 426
           " /rod_pot           14D/*, 20D, 41, 47
           " /sys_matrix_rod    64D/*, 69D, 99, 109, 111, 113, 125, 141
          smc1.f/abc_c1_invert  377, 378, 391, 392
            " /sys_matrix_c1    14D/*, 20D, 95, 117, 118, 120, 121,
                                129, 131, 133, 135, 162, 165, 190, 192,
                                201, 203, 205, 207, 237, 240, 260, 262
```

Variables :

```
Symbol      File/Subprogram       Line
------      ---- ----------       ----

k2          c1.f/output_rc1       719D, 725=, 753, 757, 778, 782, 855,
                                  928
            " /sys_matrix_rc1     263D, 269=, 276, 279, 309, 312, 377,
                                  380, 393, 412, 415, 422
            c2.f/output_rc2       1114D, 1121=, 1152, 1156, 1177, 1181,
                                  1254, 1326
            " /sys_matrix_rc2     325D, 335=, 344, 348, 391, 394, 452,
                                  455, 468, 471, 497, 500, 511, 514, 671,
                                  674, 713, 716
            rf.f/output           285D, 290=, 313, 326, 329, 354, 392
            " /sys_matrix_rod     94D, 99=, 104, 115
            smc1.f/sys_matrix_c1  87D, 95=, 102, 106, 137, 140, 179, 182,
                                  209, 212
k2_sp_cy1_a2 c1.f/output_rc1      857, 930
k2_sp_cy2_b2 c2.f/output_rc2      1256, 1328
kn          mr2cf.f/MAIN          8D
l           cbessl.f/cbessk       246D, 277=, 279
            "  /cbessy            169D, 205=, 207
l2gc1       c1.f/output_rc1       721D, 731=, 751, 755
            " /sys_matrix_rc1     262D, 270=, 275, 278, 376, 379
            c2.f/output_rc2       1117D, 1127=
            " /sys_matrix_rc2     324D, 336=, 467, 470, 670, 673
            smc1.f/sys_matrix_c1  86D, 96=, 101, 105, 178, 181
l2gc2       c2.f/output_rc2       1117D, 1128=, 1150, 1154
            " /sys_matrix_rc2     324D, 337=, 343, 347, 451, 454
l2gr        c1.f/sys_matrix_rc1   262D, 271=, 392
lame_c1     c1.f/abc_rc1_invert   559D, 574D
            " /c1a_pot            40D, 56D
            " /c1b_pot            126D, 142D
            " /output_rc1         674D, 689D, 731, 751, 752, 755, 756,
                                  777, 778, 778, 781, 782, 782
            " /sys_matrix_rc1     217D, 233D, 270, 275, 276, 278, 279,
                                  376, 377, 379, 380
            c2.f/abc_rc2_invert   826D, 841D
            " /output_rc2         1016D, 1031D, 1127
            " /sys_matrix_rc2     224D, 240D, 336, 467, 468, 470, 471,
                                  670, 671, 673, 674
            mr2cf.f/MAIN          107D, 122D, 210=, 212
            smc1.f/abc_c1_invert  305D, 320D
            " /sys_matrix_c1      40D, 56D, 96, 101, 102, 105, 106, 178,
                                  179, 181, 182
```

Variables :

Symbol	File/Subprogram	Line
lame_c2	c2.f/abc_rc2_invert	864D, 881D
	" /c2b_pot	43D, 59D
	" /c2c_pot	134D, 151D
	" /output_rc2	1054D, 1071D, 1128, 1150, 1151, 1154, 1155, 1176, 1177, 1177, 1180, 1181, 1181
	" /sys_matrix_rc2	265D, 282D, 337, 343, 344, 347, 348, 451, 452, 454, 455
	mr2cf.f/MAIN	147D, 163D, 233=, 235
lame_rod	c1.f/abc_rc1_invert	583D, 589D
	" /output_rc1	698D, 704D, 1006, 1006
	" /sys_matrix_rc1	242D, 248D, 271, 392, 393
	c2.f/abc_rc2_invert	892D, 898D
	" /output_rc2	1081D, 1087D, 1404, 1404
	" /sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D, 190=, 192
	rf.f/abc_rod_invert	211D, 217D
	" /abc_rod_solve	161D, 167D
	" /output	274D, 280D, 314, 328, 432, 432
	" /rod_pot	25D, 30D
	" /sys_matrix_rod	74D, 80D, 102, 103, 104
limit	cbessl.f/cbessi	143D, 150=, 153
	" /cbessj	60D, 81=, 86
	" /cbessk	246D, 263=, 275
	" /cbessy	169D, 191=, 203
m	" /cbessk	246D, 278=, 279
	" /cbessy	169D, 206=, 207
m_of	c1.f/abc_rc1_invert	595D, 597D, 627=, 638=
	" /output_rc1	710D, 712D
	" /sys_matrix_rc1	254D, 256D
	c2.f/abc_rc2_invert	904D, 906D, 952=, 973=
	" /output_rc2	1093D, 1095D
	" /sys_matrix_rc2	305D, 307D
	fluids.f/ofl_pot	61D, 63D
	" /output_of	145D, 147D, 158, 165
	mr2cf.f/MAIN	82D, 84D, 334, 367, 404, 435, 458
	rf.f/abc_rod_invert	223D, 225D, 248=, 253=
	" /sys_matrix_rod	86D, 88D
	smc1.f/abc_c1_invert	330D, 332D, 373=, 387=
	" /sys_matrix_c1	66D, 68D

Variables :

Symbol	File/Subprogram	Line
mains	mr2cf.f/MAIN	8D, 284=
maxpiv	rf.f/minv	460D, 479=, 483, 485=, 495
msg	mr2cf.f/MAIN	10D, 286=, 288, 488
mu_1cyl	" / "	25D, 201=, 210, 210, 210, 211
mu_2cyl	" / "	26D, 224=, 233, 233, 233, 234
mu_rod	" / "	20D, 183=, 190, 190, 190, 191
n	c1.f/abc_rc1_invert	533D/*, 538D
	" /c1a_pot	14D/*, 19D, 72, 73, 74, 75, 76, 77, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92
	" /c1b_pot	100D/*, 105D, 158, 159, 160, 161, 162, 163, 167, 168, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179
	" /output_rc1	649D/*, 653D, 723, 759, 762, 786, 789, 804, 807, 817, 819, 840, 842, 860, 863, 881, 881, 883, 883, 885, 887, 904, 906, 933, 936, 948, 950, 974, 974, 976, 976, 978, 980, 992, 994
	" /sys_matrix_rc1	191D/*, 196D, 268, 281, 284, 305, 307, 309, 310, 312, 313, 325, 328, 338, 341, 358, 360, 382, 385, 395, 408, 410, 412, 413, 415, 416, 420, 422, 423, 429, 432, 442, 445, 447, 453, 463, 465, 473, 481, 483, 493, 511, 514, 521
	c2.f/abc_rc2_invert	800D/*, 805D, 954, 955, 975, 976
	" /c2b_pot	17D/*, 22D, 76, 77, 78, 79, 80, 81, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96
	" /c2c_pot	107D/*, 112D, 168, 169, 170, 171, 172, 173, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189
	" /output_rc2	991D/*, 995D, 1119, 1158, 1161, 1185, 1188, 1203, 1206, 1216, 1218, 1239, 1241, 1259, 1262, 1279, 1279, 1281, 1281, 1283, 1285, 1302, 1304, 1331, 1334, 1346, 1348, 1372, 1372, 1374, 1374, 1376, 1378, 1390, 1392

Variables :

Symbol   File/Subprogram   Line
------   ---- ----------   ----

	" /sys_matrix_rc2	198D/*, 203D, 334, 351, 353, 355, 357, 387, 389, 391, 392, 394, 395, 415, 418, 428, 431, 457, 460, 473, 476, 493, 495, 497, 498, 500, 501, 507, 509, 511, 512, 514, 515, 523, 526, 536, 539, 541, 544, 554, 557, 569, 571, 581, 583, 595, 597, 607, 609, 633, 636, 647, 649, 676, 679, 709, 711, 713, 714, 716, 717, 738, 741, 751, 754, 779, 781
	cbessl.f/cbessh1	483D/*, 486D, 489, 489
	" /cbessh2	497D/*, 499D, 503, 503
	" /cbessi	141D/*, 143D, 149
	" /cbessj	58D/*, 60D, 71, 75, 93, 126, 127
	" /cbessk	244D/*, 246D, 255
	" /cbessy	167D/*, 169D, 179, 216, 225, 225, 226, 226, 226, 227, 229, 229, 229, 230, 232, 233
	" /d1cbessh1	511D/*, 512D, 517, 518
	" /d1cbessh2	526D/*, 527D, 532, 533
	" /d1cbessi	453D/*, 457D, 460, 460, 460
	" /d1cbessj	323D/*, 327D, 330, 330, 330
	" /d1cbessk	293D/*, 297D, 300, 300, 300
	" /d1cbessy	356D/*, 360D, 365, 369, 374, 382, 386
	" /d2cbessi	468D/*, 472D
	" /d2cbessj	338D/*, 342D, 346, 347, 347
	" /d2cbessk	308D/*, 312D
	" /d2cbessy	401D/*, 405D, 411, 416, 423, 433, 438
	" /fac	21D/*, 22D, 23, 23, 26, 27, 28
	" /gamma	4D/*, 5D, 6, 6, 9, 10, 11
	" /psi	38D/*, 39D, 43
	fluids.f/ifl_pot	14D/*, 18D, 42, 43
	" /ofl_pot	52D/*, 54D, 73, 74, 79, 80
	" /output_if	91D/*, 95D
	" /output_of	134D/*, 138D

Variables :

Symbol	File/Subprogram	Line
	mr2cf.f/MAIN	19D, 268=, 299, 310=, 311=, 312=, 313=, 314=, 316=, 321=, 322=, 326=, 327=, 332=, 338=, 353=, 354=, 355=, 360=, 361=, 366=, 368=, 382=, 383=, 384=, 385=, 386=, 388=, 392=, 393=, 397=, 398=, 403=, 406=, 421=, 422=, 423=, 424=, 425=, 430=, 431=, 433=, 440=, 441=, 445=, 446=, 450=, 451=, 456=, 460=
	rf.f/abc_rod_invert	231D, 234=, 237=
	" /minv	448D/*, 455D, 456, 456, 456, 456, 456, 456, 463, 464, 469, 470, 471, 472, 481, 489, 500, 501, 510, 510, 512, 514, 519, 522, 528, 529, 530
	" /output	264D/*, 268D, 288, 315, 316, 330, 337, 342, 348, 356, 367, 367, 368, 378, 379, 394, 401, 402, 416, 416, 417, 424, 425
	" /rod_pot	14D/*, 19D, 42, 43, 44, 48, 49, 50, 52, 53, 54
	" /sys_matrix_rod	64D/*, 68D, 98, 106, 107, 113, 115, 116, 118, 118, 125, 139
	smc1.f/abc_c1_invert	279D/*, 284D, 375, 376, 389, 390
	" /sys_matrix_c1	14D/*, 19D, 94, 109, 111, 113, 115, 133, 135, 137, 138, 140, 141, 150, 153, 163, 166, 184, 187, 205, 207, 209, 210, 212, 213, 225, 228, 238, 241, 256, 258
n2	c1.f/output_rc1	718D, 723=, 752, 756, 778, 782, 877, 879, 970, 972
	" /sys_matrix_rc1	264D, 268=, 276, 279, 332, 335, 377, 380, 393, 436, 439, 450
	c2.f/output_rc2	1113D, 1119=, 1151, 1155, 1177, 1181, 1275, 1277, 1368, 1370
	" /sys_matrix_rc2	326D, 334=, 344, 348, 422, 425, 452, 455, 468, 471, 530, 533, 548, 551, 671, 674, 745, 748
	rf.f/output	284D, 288=, 313, 326, 366, 415
	" /sys_matrix_rod	95D, 98=, 104, 122
	smc1.f/sys_matrix_c1	88D, 94=, 102, 106, 157, 160, 179, 182, 232, 235

Variables :

Symbol	File/Subprogram	Line
na	cbessl.f/cbessi	143D, 149=, 154, 158
	" /cbessj	60D, 71=, 87, 91, 94
	" /cbessk	246D, 255=, 265, 266, 271, 273, 273, 277, 278, 283, 283
	" /cbessy	169D, 179=, 193, 194, 199, 201, 205, 206, 211, 217
	" /psi	39D, 43=, 44, 47
no	c1.f/output_rc1	719D, 728=, 962, 1007
	c2.f/output_rc2	1114D, 1124=, 1360, 1405
	rf.f/output	285D, 293=, 409, 433
ofluid	c1.f/abc_rc1_invert	597D
	" /output_rc1	710D
	" /sys_matrix_rc1	254D
	c2.f/abc_rc2_invert	906D
	" /output_rc2	1095D
	" /sys_matrix_rc2	305D
	fluids.f/ofl_pot	61D
	" /output_of	145D
	mr2cf.f/MAIN	84D
	rf.f/abc_rod_invert	225D
	" /sys_matrix_rod	86D
	smc1.f/abc_c1_invert	332D
	" /sys_matrix_c1	66D
ofsc	c1.f/abc_rc1_invert	595D, 597D
	" /output_rc1	710D, 712D
	" /sys_matrix_rc1	254D, 256D, 297
	c2.f/abc_rc2_invert	904D, 906D
	c2.f/output_rc2	1093D, 1095D
	" /sys_matrix_rc2	305D, 307D, 345, 349, 353, 357, 360, 363
	fluids.f/ofl_pot	61D, 63D, 74=, 80=
	" /output_of	145D, 147D, 158
	mr2cf.f/MAIN	82D, 84D, 334, 367, 404, 435, 458
	rf.f/abc_rod_invert	223D, 225D
	" /sys_matrix_rod	86D, 88D, 130
	smc1.f/abc_c1_invert	330D, 332D
	" /sys_matrix_c1	66D, 68D, 103, 107, 111, 115, 118, 121

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
om	c1.f/c1a_pot	14D/*, 20D, 71, 79
	" /c1b_pot	100D/*, 106D, 157, 166
	" /sys_matrix_rc1	192D/*, 197D, 297
	c2.f/c2b_pot	17D/*, 23D, 75, 83
	" /c2c_pot	107D/*, 113D, 167, 176
	" /sys_matrix_rc2	199D/*, 204D, 345, 349, 353, 357, 360, 363, 686
	fluids.f/ifl_pot	14D/*, 19D, 39
	" /ofl_pot	52D/*, 55D, 70, 72, 78
	" /output_if	91D/*, 96D, 116, 123
	" /output_of	134D/*, 139D, 158, 165
	mr2cf.f/MAIN	21D, 265=, 284, 297=, 310=, 311=, 312=, 313=, 315=, 321=, 322=, 326=, 332=, 338=, 353=, 354=, 355=, 360=, 366=, 368=, 382=, 383=, 384=, 385=, 386=, 392=, 397=, 403=, 406=, 421=, 422=, 423=, 424=, 425=, 430=, 431=, 440=, 441=, 445=, 450=, 456=, 460=, 480
	rf.f/rod_pot	14D/*, 20D, 41, 47
	" /sys_matrix_rod	64D/*, 69D, 130
	smc1.f/sys_matrix_c1	15D/*, 20D, 103, 107, 111, 115, 118, 121, 194
p	c1.f/c1a_pot	67D, 71=, 72, 73, 74, 75, 76, 77
	" /c1b_pot	153D, 157=, 158, 159, 160, 161, 162, 163
	c2.f/c2b_pot	71D, 75=, 76, 77, 78, 79, 80, 81
	" /c2c_pot	163D, 167=, 168, 169, 170, 171, 172, 173
	rf.f/rod_pot	38D, 41=, 42, 43, 44
p11	c1.f/output_rc1	719D, 729=, 962, 1007
	c2.f/output_rc2	1114D, 1125=, 1360, 1405
	rf.f/output	285D, 294=, 409, 433
p12	c1.f/output_rc1	719D, 730=, 962, 962, 1007, 1007
	c2.f/output_rc2	1114D, 1126=, 1360, 1360, 1405, 1405
	rf.f/output	285D, 295=, 409, 409, 433, 433
part1	cbessl.f/cbessk	248D, 258=, 271=, 285
	" /cbessy	171D, 186=, 199=, 213
part2	" /cbessk	248D, 259=, 273=, 285
	" /cbessy	171D, 187=, 201=, 213
part3	" /cbessk	248D, 260=, 283=, 285
	" /cbessy	171D, 188=, 211=, 213

Variables :

Symbol	File/Subprogram	Line
pi	" /cbessj	61D, 68D, 126, 126, 127, 127, 127
	" /cbessk	247D, 251D
	" /cbessy	170D, 175D, 199, 201, 211, 232, 232, 232, 233, 233
	fluids.f/output_if	111D, 116=, 118
ps	" /output_of	153D, 158=, 160
psi	cbessl.f/cbessk	247D
	" /cbessy	170D
	" /psi	40D, 45=, 50=
	mr2cf.f/MAIN	33D
q	c1.f/c1a_pot	67D, 79=, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92
	" /c1b_pot	153D, 166=, 167, 168, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179
	c2.f/c2b_pot	71D, 83=, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96
	" /c2c_pot	163D, 176=, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189
	rf.f/rod_pot	38D, 47=, 48, 49, 50, 52, 53, 54
r	c1.f/c1a_pot	14D/*, 20D, 72, 73, 74, 75, 76, 77, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92
	" /c1b_pot	100D/*, 106D, 158, 159, 160, 161, 162, 163, 167, 168, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179
	" /output_rc1	649D/*, 654D, 724, 752, 756, 759, 760, 762, 763, 777, 781, 786, 789, 804, 807, 817, 819, 840, 842, 860, 863, 877, 879, 881, 883, 885, 887, 904, 904, 906, 906, 933, 936, 948, 948, 950, 950, 970, 972, 974, 976, 978, 980, 992, 992, 994, 994
	c2.f/c2b_pot	17D/*, 23D, 76, 77, 78, 79, 80, 81, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96
	" /c2c_pot	107D/*, 113D, 168, 169, 170, 171, 172, 173, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189

Variables :

Symbol   File/Subprogram   Line
------   ---- ----------   ----

```
          " /output_rc2      991D/*, 996D, 1120, 1151, 1155, 1158,
                             1159, 1161, 1162, 1176, 1180, 1185,
                             1188, 1203, 1206, 1216, 1218, 1239,
                             1241, 1259, 1262, 1275, 1277, 1279,
                             1281, 1283, 1285, 1302, 1302, 1304,
                             1304, 1331, 1334, 1346, 1346, 1348,
                             1348, 1368, 1370, 1372, 1374, 1376,
                             1378, 1390, 1390, 1392, 1392
          cbessl.f/cbessh1   483D/*, 484D, 489, 489
           "  /cbessh2       497D/*, 498D, 503, 503
           "  /cbessi        141D/*, 144D, 148
           "  /cbessj        58D/*, 61D, 70
           "  /cbessk        244D/*, 247D, 253, 273
           "  /cbessy        167D/*, 170D, 177, 201
           "  /d1cbessh1     511D/*, 513D, 517, 518
           "  /d1cbessh2     526D/*, 528D, 532, 533
           "  /d1cbessi      453D/*, 455D, 460, 460, 460
           "  /d1cbessj      323D/*, 325D, 330, 330, 330
           "  /d1cbessk      293D/*, 295D, 300, 300, 300
           "  /d1cbessy      356D/*, 358D, 367, 371, 371, 371, 372,
                             372, 376, 376, 376, 377, 377, 378, 378,
                             379, 384, 384, 384, 388, 388, 388, 389,
                             389
           "  /d2cbessi      468D/*, 470D
           "  /d2cbessj      338D/*, 340D, 346, 347, 347
           "  /d2cbessk      308D/*, 310D
           "  /d2cbessy      401D/*, 403D, 413, 413, 413, 414, 414,
                             418, 418, 418, 419, 419, 420, 420, 421,
                             425, 425, 425, 426, 426, 427, 427, 428,
                             428, 429, 429, 429, 430, 430, 435, 435,
                             435, 436, 436, 440, 440, 440, 441, 441,
                             441, 442
          fluids.f/ifl_pot   14D/*, 19D, 42, 43
           "  /ofl_pot       52D/*, 55D, 73, 74, 79, 80
           "  /output_if     91D/*, 96D
           "  /output_of     134D/*, 139D
          mr2cf.f/MAIN       20D, 267=, 299, 318, 321=, 322=, 324,
                             324, 326=, 327=, 330, 332=, 338=, 358,
                             360=, 361=, 364, 366=, 368=, 390, 392=,
                             393=, 395, 395, 397=, 398=, 401, 403=,
```

Variables :

Symbol	File/Subprogram	Line
		406=, 437, 440=, 441=, 443, 443, 445=, 446=, 448, 448, 450=, 451=, 454, 456=, 460=
	rf.f/output	264D/*, 269D, 289, 312, 316, 325, 330, 337, 337, 342, 348, 356, 366, 367, 368, 379, 394, 402, 415, 416, 417, 425
	" /rod_pot	14D/*, 20D, 42, 43, 44, 48, 49, 50, 52, 53, 54
r2	c1.f/output_rc1	719D, 724=, 753, 757, 778, 782, 877, 879, 881, 883, 970, 972, 974, 976
	c2.f/output_rc2	1114D, 1120=, 1152, 1156, 1177, 1181, 1275, 1277, 1279, 1281, 1368, 1370, 1372, 1374
	rf.f/output	285D, 289=, 313, 315, 326, 366, 367, 378, 401, 415, 416, 424
r_1cyl	mr2cf.f/MAIN	25D, 203=, 212, 213
r_2cyl	" / "	26D, 226=, 235, 236
r_rod	" / "	20D, 184=, 192, 193
ri	c2.f/sys_matrix_rc2	199D/*, 204D, 686
	fluids.f/output_if	91D/*, 96D, 116
	mr2cf.f/MAIN	23D, 278=, 315=, 322=, 432=, 441=
	smc1.f/sys_matrix_c1	15D/*, 20D, 194
ro	c1.f/sys_matrix_rc1	192D/*, 197D, 297
	c2.f/sys_matrix_rc2	199D/*, 204D, 345, 349, 353, 357, 360, 363
	fluids.f/output_of	134D/*, 139D, 158
	mr2cf.f/MAIN	21D, 275=, 315=, 338=, 355=, 368=, 386=, 406=, 431=, 460=
	rf.f/sys_matrix_rod	64D/*, 69D, 130
	smc1.f/sys_matrix_c1	15D/*, 20D, 103, 107, 111, 115, 118, 121

Variables :

Symbol	File/Subprogram	Line
rod	c1.f/abc_rc1_invert	581D
	" /output_rc1	696D
	" /sys_matrix_rc1	240D
	c2.f/abc_rc2_invert	890D
	" /output_rc2	1079D
	" /sys_matrix_rc2	291D
	mr2cf.f/MAIN	72D
	rf.f/abc_rod_invert	209D
	" /abc_rod_solve	159D
	" /output	272D
	" /rod_pot	23D
	" /sys_matrix_rod	72D
s1	" /minv	460D, 482=, 483
shear_c1	c1.f/abc_rc1_invert	559D, 574D
	" /c1a_pot	40D, 56D
	" /c1b_pot	126D, 142D
	" /output_rc1	674D, 689D, 731, 759, 762, 765, 767, 779, 783, 786, 789
	" /sys_matrix_rc1	217D, 233D, 270, 281, 284, 287, 289, 301, 303, 305, 307, 309, 312, 325, 328, 331, 334, 337, 340, 382, 385, 388, 390, 404, 406, 408, 410, 412, 415, 429, 432, 435, 438, 441, 444
	c2.f/abc_rc2_invert	826D, 841D
	" /output_rc2	1016D, 1031D, 1127
	" /sys_matrix_rc2	224D, 240D, 336, 473, 476, 479, 481, 503, 505, 507, 509, 511, 514, 541, 544, 547, 550, 553, 556, 676, 679, 682, 684, 705, 707, 709, 711, 713, 716, 738, 741, 744, 747, 750, 753
	mr2cf.f/MAIN	107D, 122D, 211=, 212, 213
	smc1.f/abc_c1_invert	305D, 320D
	" /sys_matrix_c1	40D, 56D, 96, 109, 113, 117, 120, 129, 131, 133, 135, 137, 140, 150, 153, 156, 159, 162, 165, 184, 187, 190, 192, 201, 203, 205, 207, 209, 212, 225, 228, 231, 234, 237, 240

Variables :

```
Symbol     File/Subprogram       Line
------     ---- ----------       ----

shear_c2   c2.f/abc_rc2_invert   864D, 881D
           " /c2b_pot            43D, 59D
           " /c2c_pot            134D, 151D
           " /output_rc2         1054D, 1071D, 1128, 1158, 1161, 1164,
                                 1166, 1178, 1182, 1185, 1188
           " /sys_matrix_rc2     265D, 282D, 337, 351, 355, 359, 362,
                                 383, 385, 387, 389, 391, 394, 415, 418,
                                 421, 424, 427, 430, 457, 460, 463, 465,
                                 489, 491, 493, 495, 497, 500, 523, 526,
                                 529, 532, 535, 538
           mr2cf.f/MAIN          147D, 163D, 234=, 235, 236
shear_rod  c1.f/abc_rc1_invert   583D, 589D
           " /output_rc1         698D, 704D, 1006
           " /sys_matrix_rc1     242D, 248D, 271, 395, 398, 418, 420,
                                 422, 447, 449, 452
           c2.f/abc_rc2_invert   892D, 898D
           " /output_rc2         1081D, 1087D, 1404
           " /sys_matrix_rc2     293D, 299D
           mr2cf.f/MAIN          70D, 74D, 191=, 192, 193
           rf.f/abc_rod_invert   211D, 217D
           " /abc_rod_solve      161D, 167D
           " /output             274D, 280D, 314, 328, 432
           " /rod_pot            25D, 30D
           " /sys_matrix_rod     74D, 80D, 102, 106, 107, 109, 111, 113,
                                 115, 118, 121, 125
si         mr2cf.f/MAIN          9D
size       c1.f/abc_rc1_invert   603D, 606=, 610=
           c2.f/abc_rc2_invert   924D, 927=, 931=
           mr2cf.f/MAIN          32D, 281=
           smc1.f/abc_c1_invert  350D, 353=, 357=
sm         mr2cf.f/MAIN          34D, 355=, 356=
           rf.f/abc_rod_invert   201D/*, 207D, 237=
           " /abc_rod_solve      152D/*, 157D, 177, 177, 177, 177, 178,
                                 178, 178, 178, 179, 179, 179, 179, 181,
                                 181, 181, 181, 182, 182, 182, 182, 183,
                                 183, 183, 183, 186, 186, 186, 187, 187,
                                 187, 187, 187, 187, 188, 188, 188, 188,
                                 188, 188, 189, 189, 189
           " /sys_matrix_rod     64D/*, 70D, 102=, 106=, 109=, 111=,
                                 113=, 115=, 118=, 121=, 124=, 130=,
                                 133=, 135=, 137=, 139=, 141=, 143=
```

Variables :

```
Symbol    File/Subprogram       Line
------    ---- ----------       ----

smc1      mr2cf.f/MAIN          59D, 315=, 316=
          smc1.f/abc_c1_invert  279D/*, 286D, 357=
            " /sys_matrix_c1    15D/*, 21D, 101=, 105=, 109=, 113=,
                                117=, 120=, 123=, 129=, 131=, 133=,
                                135=, 137=, 140=, 143=, 150=, 153=,
                                156=, 159=, 162=, 165=, 168=, 178=,
                                181=, 184=, 187=, 190=, 192=, 194=,
                                201=, 203=, 205=, 207=, 209=, 212=,
                                215=, 225=, 228=, 231=, 234=, 237=,
                                240=, 243=, 252=, 254=, 256=, 258=,
                                260=, 262=, 264=
smc1inv   mr2cf.f/MAIN          59D
          smc1.f/abc_c1_invert  351D, 357=, 367, 368, 369, 370, 371,
                                372, 379, 381, 382, 383, 384, 385, 386,
                                393
sminv     mr2cf.f/MAIN          35D
          rf.f/abc_rod_invert   232D, 237=, 245, 246, 247, 248, 250,
                                251, 252, 253
smrc1     c1.f/abc_rc1_invert   533D/*, 540D, 610=
          c1.f/sys_matrix_rc1   192D/*, 198D, 275=, 278=, 281=, 284=,
                                287=, 289=, 291=, 293=, 295=, 297=,
                                301=, 303=, 305=, 307=, 309=, 312=,
                                315=, 317=, 319=, 321=, 325=, 328=,
                                331=, 334=, 337=, 340=, 343=, 345=,
                                347=, 349=, 354=, 356=, 358=, 360=,
                                362=, 364=, 366=, 368=, 370=, 372=,
                                376=, 379=, 382=, 385=, 388=, 390=,
                                392=, 395=, 398=, 400=, 404=, 406=,
                                408=, 410=, 412=, 415=, 418=, 420=,
                                422=, 425=, 429=, 432=, 435=, 438=,
                                441=, 444=, 447=, 449=, 452=, 455=,
                                459=, 461=, 463=, 465=, 467=, 469=,
                                471=, 473=, 475=, 477=, 481=, 483=,
                                485=, 487=, 489=, 491=, 493=, 495=,
                                497=, 499=, 503=, 505=, 507=, 509=,
                                511=, 514=, 517=, 519=, 521=, 524=
          mr2cf.f/MAIN          48D, 387=, 388=
smrc1inv  c1.f/abc_rc1_invert   604D, 610=, 618, 619, 620, 621, 622,
                                623, 624, 625, 626, 627, 629, 630, 631,
                                632, 633, 634, 635, 636, 637, 638
          mr2cf.f/MAIN          48D
```

Variables :

```
Symbol     File/Subprogram        Line
------     ---- ----------        ----

smrc2     c2.f/abc_rc2_invert     800D/*, 807D, 931=
          " /sys_matrix_rc2       199D/*, 205D, 343=, 347=, 351=, 355=,
                                  359=, 362=, 365=, 367=, 369=, 371=,
                                  373=, 375=, 377=, 383=, 385=, 387=,
                                  389=, 391=, 394=, 397=, 399=, 401=,
                                  403=, 405=, 407=, 409=, 415=, 418=,
                                  421=, 424=, 427=, 430=, 433=, 435=,
                                  437=, 439=, 441=, 443=, 445=, 451=,
                                  454=, 457=, 460=, 463=, 465=, 467=,
                                  470=, 473=, 476=, 479=, 481=, 483=,
                                  489=, 491=, 493=, 495=, 497=, 500=,
                                  503=, 505=, 507=, 509=, 511=, 514=,
                                  517=, 523=, 526=, 529=, 532=, 535=,
                                  538=, 541=, 544=, 547=, 550=, 553=,
                                  556=, 559=, 565=, 567=, 569=, 571=,
                                  573=, 575=, 577=, 579=, 581=, 583=,
                                  585=, 587=, 589=, 595=, 597=, 599=,
                                  601=, 603=, 605=, 607=, 609=, 611=,
                                  613=, 615=, 617=, 619=, 625=, 627=,
                                  629=, 631=, 633=, 636=, 639=, 641=,
                                  643=, 645=, 647=, 649=, 651=, 658=,
                                  660=, 662=, 664=, 666=, 668=, 670=,
                                  673=, 676=, 679=, 682=, 684=, 686=,
                                  693=, 695=, 697=, 699=, 701=, 703=,
                                  705=, 707=, 709=, 711=, 713=, 716=,
                                  719=, 726=, 728=, 730=, 732=, 734=,
                                  736=, 738=, 741=, 744=, 747=, 750=,
                                  753=, 763=, 765=, 767=, 769=, 771=,
                                  773=, 775=, 777=, 779=, 781=, 783=,
                                  785=, 787=
          mr2cf.f/MAIN            54D, 432=, 433=
smrc2inv  c2.f/abc_rc2_invert     925D, 931=, 939, 940, 941, 942, 943,
                                  944, 945, 946, 947, 948, 949, 950, 951,
                                  960, 961, 962, 963, 964, 965, 966, 967,
                                  968, 969, 970, 971, 972
          mr2cf.f/MAIN            54D
```

Variables :

```
Symbol     File/Subprogram        Line
------     ---- ----------        ----

sp_cy1_a1  c1.f/abc_rc1_invert    544D, 562D
           " /c1a_pot             25D, 44D, 74=
           " /c1b_pot             111D, 130D
           " /output_rc1          659D, 677D, 752, 778, 799, 817, 855,
                                  877, 928, 970
           " /sys_matrix_rc1      202D, 221D, 377, 429, 481, 503
           c2.f/abc_rc2_invert    811D, 829D
           " /output_rc2          1001D, 1019D
           " /sys_matrix_rc2      209D, 228D, 671, 739
           mr2cf.f/MAIN           92D, 110D
           smc1.f/abc_c1_invert   290D, 308D
           " /sys_matrix_c1       25D, 44D, 179, 226
sp_cy1_a2  c1.f/abc_rc1_invert    545D, 563D
           " /c1a_pot             26D, 45D, 77=
           " /c1b_pot             112D, 131D
           " /output_rc1          660D, 678D, 756, 782, 801, 819, 879,
                                  972
           " /sys_matrix_rc1      203D, 222D, 380, 432, 483, 505
           c2.f/abc_rc2_invert    812D, 830D
           " /output_rc2          1002D, 1020D
           " /sys_matrix_rc2      210D, 229D, 674, 742
           mr2cf.f/MAIN           93D, 111D
           smc1.f/abc_c1_invert   291D, 309D
           " /sys_matrix_c1       26D, 45D, 182, 229
sp_cy1_b1  c1.f/abc_rc1_invert    546D, 564D
           " /c1a_pot             27D, 46D
           " /c1b_pot             113D, 132D, 160=
           " /output_rc1          661D, 679D
           " /sys_matrix_rc1      204D, 223D, 276, 326
           c2.f/abc_rc2_invert    813D, 831D
           " /output_rc2          1003D, 1021D
           " /sys_matrix_rc2      211D, 230D, 468, 541, 607, 639
           mr2cf.f/MAIN           94D, 112D
           smc1.f/abc_c1_invert   292D, 310D
           " /sys_matrix_c1       27D, 46D, 102, 151
```

Variables :

```
Symbol     File/Subprogram        Line
------     ---- ----------        ----

sp_cy1_b2  c1.f/abc_rc1_invert    547D, 565D
             " /c1a_pot           28D, 47D
             " /c1b_pot           114D, 133D, 163=
             " /output_rc1        662D, 680D
             " /sys_matrix_rc1    205D, 224D, 279, 329
           c2.f/abc_rc2_invert    814D, 832D
             " /output_rc2        1004D, 1022D
             " /sys_matrix_rc2    212D, 231D, 471, 544, 609, 641
           mr2cf.f/MAIN           95D, 113D
           smc1.f/abc_c1_invert   293D, 311D
             " /sys_matrix_c1     28D, 47D, 106, 154
sp_cy2_b1  c2.f/abc_rc2_invert    849D, 869D
             " /c2b_pot           28D, 47D, 78=
             " /c2c_pot           119D, 139D
             " /output_rc2        1039D, 1059D, 1151, 1177, 1198, 1216,
                                  1254, 1275, 1326, 1368
             " /sys_matrix_rc2    250D, 270D, 452, 523, 595, 625
           mr2cf.f/MAIN           132D, 151D
sp_cy2_b2  c2.f/abc_rc2_invert    850D, 870D
             " /c2b_pot           29D, 48D, 81=
             " /c2c_pot           120D, 140D
             " /output_rc2        1040D, 1060D, 1155, 1181, 1200, 1218,
                                  1277, 1370
             " /sys_matrix_rc2    251D, 271D, 455, 526, 597, 627
           mr2cf.f/MAIN           133D, 152D
sp_cy2_c1  c2.f/abc_rc2_invert    851D, 871D
             " /c2b_pot           30D, 49D
             " /c2c_pot           121D, 141D, 170=
             " /output_rc2        1041D, 1061D
             " /sys_matrix_rc2    252D, 272D, 344, 416
           mr2cf.f/MAIN           134D, 153D
sp_cy2_c2  c2.f/abc_rc2_invert    852D, 872D
             " /c2b_pot           31D, 50D
             " /c2c_pot           122D, 142D, 173=
             " /output_rc2        1042D, 1062D
           c2.f/sys_matrix_rc2    253D, 273D, 348, 419
           mr2cf.f/MAIN           135D, 154D
```

Variables :

Symbol	File/Subprogram	Line
sp_rod	c1.f/abc_rc1_invert	581D, 586D
	" /output_rc1	696D, 701D
	" /sys_matrix_rc1	240D, 245D, 393, 447, 493, 517
	c2.f/abc_rc2_invert	890D, 895D
	" /output_rc2	1079D, 1084D
	" /sys_matrix_rc2	291D, 296D
	mr2cf.f/MAIN	67D, 72D
	rf.f/abc_rod_invert	209D, 214D
	" /abc_rod_solve	159D, 164D
	" /output	272D, 277D, 313, 313, 326, 326, 329, 336, 342, 354, 366, 392, 415
	" /rod_pot	23D, 27D, 44=
	" /sys_matrix_rod	72D, 77D, 104, 118
srr	c1.f/output_rc1	720D, 769=, 771
	c2.f/output_rc2	1115D, 1168=, 1170
	rf.f/output	286D, 314=, 319
sum	cbessl.f/cbessi	145D, 155=, 156
	" /cbessj	62D, 88=, 89
	" /fac	22D, 26=, 28, 28=, 30
	" /gamma	5D, 9=, 11, 11=, 13
	" /psi	40D, 42=, 48, 48=, 50
sum1	" /cbessk	248D, 267=, 268
	" /cbessy	171D, 195=, 196
sum2	" /cbessk	248D, 279=, 280
	" /cbessy	171D, 207=, 208
switch	rf.f/minv	460D, 490=, 492
sxx	c1.f/output_rc1	720D, 791=, 793
	c2.f/output_rc2	1115D, 1190=, 1192
	rf.f/output	286D, 328=, 332
t	" /minv	461D, 535=
tft	c1.f/output_rc1	649D/*, 653D, 749, 775, 797, 814, 834, 853, 875, 898, 921, 926, 968
	c2.f/output_rc2	991D/*, 995D, 1148, 1174, 1196, 1213, 1233, 1252, 1273, 1296, 1319, 1324, 1366
	fluids.f/output_if	91D/*, 95D, 114, 121
	" /output_of	134D/*, 138D, 156, 163
	mr2cf.f/MAIN	19D, 269=, 294, 299, 322=, 327=, 338=, 342, 361=, 368=, 372, 393=, 398=, 406=, 410, 441=, 446=, 451=, 460=, 464

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
	rf.f/output	264D/*, 268D, 310, 323, 334, 340, 346, 352, 364, 375, 388, 390, 413
total	cbessl.f/cbessi	145D, 151=, 156, 156=, 158
	" /cbessj	62D, 82=, 89, 89=, 91
total1	" /cbessk	248D, 261=, 268, 268=, 271
	" /cbessy	172D, 189=, 196, 196=, 199
total2	" /cbessk	248D, 262=, 280, 280=, 283
	" /cbessy	172D, 190=, 208, 208=, 211
uc	c1.f/output_rc1	720D, 848=, 850
	c2.f/output_rc2	1115D, 1247=, 1249
ur	rf.f/output	286D, 348=, 350
value	c1.f/output_rc1	649D/*, 655D, 771=, 793=, 811=, 831=, 850=, 869=, 872=, 892=, 895=, 915=, 918=, 922=, 964=, 985=, 1008=
	c2.f/output_rc2	991D/*, 997D, 1170=, 1192=, 1210=, 1230=, 1249=, 1267=, 1270=, 1290=, 1293=, 1313=, 1316=, 1320=, 1362=, 1383=, 1406=
	fluids.f/output_if	91D/*, 97D, 118=, 125=
	" /output_of	134D/*, 140D, 160=, 167=
	mr2cf.f/MAIN	36D, 322=, 327=, 338=, 343, 343=, 346, 361=, 368=, 373, 373=, 376, 393=, 398=, 406=, 411, 411=, 414, 441=, 446=, 451=, 460=, 465, 465=, 468, 472, 476
	rf.f/output	264D/*, 270D, 319=, 332=, 338=, 344=, 350=, 359=, 362=, 371=, 373=, 383=, 386=, 389=, 411=, 420=, 434=
vc	c1.f/output_rc1	720D, 829=, 831
	c2.f/output_rc2	1115D, 1228=, 1230
vr	rf.f/output	286D, 342=, 344

Variables :

```
Symbol       File/Subprogram          Line
------       ---- ----------          ----

vrtp_cy1_a1  c1.f/abc_rc1_invert       554D, 570D
              " /c1a_pot               35D, 52D, 89=
              " /c1b_pot               121D, 138D
              " /output_rc1            669D, 685D, 786, 803, 825, 844, 860,
                                    885, 933, 978
              " /sys_matrix_rc1        212D, 229D, 412, 442, 467, 489, 511
             c2.f/abc_rc2_invert       821D, 837D
              " /output_rc2            1011D, 1027D
              " /sys_matrix_rc2        219D, 236D, 713, 751, 783
             mr2cf.f/MAIN              102D, 118D
             smc1.f/abc_c1_invert      300D, 316D
              " /sys_matrix_c1         35D, 52D, 209, 238, 260
vrtp_cy1_a2  c1.f/abc_rc1_invert       555D, 571D
              " /c1a_pot               36D, 53D, 92=
              " /c1b_pot               122D, 139D
              " /output_rc1            670D, 686D, 789, 806, 827, 846, 863,
                                    887, 936, 980
              " /sys_matrix_rc1        213D, 230D, 415, 445, 469, 491, 514
             c2.f/abc_rc2_invert       822D, 838D
              " /output_rc2            1012D, 1028D
              " /sys_matrix_rc2        220D, 237D, 716, 754, 785
             mr2cf.f/MAIN              103D, 119D
             smc1.f/abc_c1_invert      301D, 317D
              " /sys_matrix_c1         36D, 53D, 212, 241, 262
vrtp_cy1_b1  c1.f/abc_rc1_invert       556D, 572D
              " /c1a_pot               37D, 54D
              " /c1b_pot               123D, 140D, 176=
              " /output_rc1            671D, 687D
              " /sys_matrix_rc1        214D, 231D, 309, 338, 362
             c2.f/abc_rc2_invert       823D, 839D
              " /output_rc2            1013D, 1029D
              " /sys_matrix_rc2        221D, 238D, 511, 554, 585, 615, 647
             mr2cf.f/MAIN              104D, 120D
             smc1.f/abc_c1_invert      302D, 318D, 377, 391
              " /sys_matrix_c1         37D, 54D, 118, 137, 163
```

Variables :

```
Symbol      File/Subprogram         Line
------      ---- ----------         ----

vrtp_cy1_b2  c1.f/abc_rc1_invert     557D, 573D
             " /c1a_pot              38D, 55D
             " /c1b_pot              124D, 141D, 179=
             " /output_rc1           672D, 688D
             " /sys_matrix_rc1       215D, 232D, 312, 341, 364
            c2.f/abc_rc2_invert      824D, 840D
             " /output_rc2           1014D, 1030D
             " /sys_matrix_rc2       222D, 239D, 514, 557, 587, 617, 649
            mr2cf.f/MAIN             105D, 121D
            smc1.f/abc_c1_invert     303D, 319D, 378, 392
             " /sys_matrix_c1        38D, 55D, 121, 140, 166
vrtp_cy2_b1  c2.f/abc_rc2_invert     859D, 877D
             " /c2b_pot              38D, 55D, 93=
             " /c2c_pot              129D, 147D
             " /output_rc2           1049D, 1067D, 1185, 1202, 1224, 1243,
                                     1259, 1283, 1331, 1376
             " /sys_matrix_rc2       260D, 278D, 497, 536, 573, 603, 633
            mr2cf.f/MAIN             142D, 159D
vrtp_cy2_b2  c2.f/abc_rc2_invert     860D, 878D
             " /c2b_pot              39D, 56D, 96=
             " /c2c_pot              130D, 148D
            c2.f/output_rc2          1050D, 1068D, 1188, 1205, 1226, 1245,
                                     1262, 1285, 1334, 1378
             " /sys_matrix_rc2       261D, 279D, 500, 539, 575, 605, 636
            mr2cf.f/MAIN             143D, 160D
vrtp_cy2_c1  c2.f/abc_rc2_invert     861D, 879D, 956, 977
             " /c2b_pot              40D, 57D
             " /c2c_pot              131D, 149D, 186=
             " /output_rc2           1051D, 1069D
             " /sys_matrix_rc2       262D, 280D, 360, 391, 428
            mr2cf.f/MAIN             144D, 161D
vrtp_cy2_c2  c2.f/abc_rc2_invert     862D, 880D, 957, 978
             " /c2b_pot              41D, 58D
             " /c2c_pot              132D, 150D, 189=
             " /output_rc2           1052D, 1070D
             " /sys_matrix_rc2       263D, 281D, 363, 394, 431
            mr2cf.f/MAIN             145D, 162D
```

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

vrtp_rod    c1.f/abc_rc1_invert    582D, 588D
              " /output_rc1        697D, 703D
              " /sys_matrix_rc1    241D, 247D, 422, 453, 475, 497, 521
            c2.f/abc_rc2_invert    891D, 897D
              " /output_rc2        1080D, 1086D
              " /sys_matrix_rc2    292D, 298D
            mr2cf.f/MAIN           69D, 73D
            rf.f/abc_rod_invert    210D, 216D
              " /abc_rod_solve     160D, 166D
              " /output            273D, 279D, 330, 337, 337, 343, 349,
                                   356, 368, 394, 417
              " /rod_pot           24D, 29D, 54=
              " /sys_matrix_rod    73D, 79D, 115, 124, 141
vui         fluids.f/output_if     111D, 123=, 125
vws            "  /output_of       153D, 165=, 167
vxp_cy1_a1  c1.f/abc_rc1_invert    549D, 566D
              " /c1a_pot           30D, 48D, 82=
              " /c1b_pot           116D, 134D
              " /output_rc1        664D, 681D, 759, 840, 881, 904, 948,
                                   974, 992
              " /sys_matrix_rc1    207D, 225D, 382, 408, 436, 463
            c2.f/abc_rc2_invert    816D, 833D
              " /output_rc2        1006D, 1023D
              " /sys_matrix_rc2    214D, 232D, 676, 709, 745, 779
            mr2cf.f/MAIN           97D, 114D
            smc1.f/abc_c1_invert   295D, 312D
              " /sys_matrix_c1     30D, 48D, 184, 205, 232, 256
vxp_cy1_a2  c1.f/abc_rc1_invert    550D, 567D
              " /c1a_pot           31D, 49D, 85=
              " /c1b_pot           117D, 135D
              " /output_rc1        665D, 682D, 762, 842, 883, 906, 950,
                                   976, 994
              " /sys_matrix_rc1    208D, 226D, 385, 410, 439, 465
            c2.f/abc_rc2_invert    817D, 834D
              " /output_rc2        1007D, 1024D
              " /sys_matrix_rc2    215D, 233D, 679, 711, 748, 781
            mr2cf.f/MAIN           98D, 115D
            smc1.f/abc_c1_invert   296D, 313D
              " /sys_matrix_c1     31D, 49D, 187, 207, 235, 258
```

Variables :

```
Symbol      File/Subprogram        Line
------      ---- ----------        ----

vxp_cy1_b1   c1.f/abc_rc1_invert      551D, 568D
              " /c1a_pot              32D, 50D
              " /c1b_pot              118D, 136D, 169=
              " /output_rc1           666D, 683D
              " /sys_matrix_rc1       209D, 227D, 281, 305, 332, 358
             c2.f/abc_rc2_invert      818D, 835D
              " /output_rc2           1008D, 1025D
              " /sys_matrix_rc2       216D, 234D, 473, 507, 548, 581
             mr2cf.f/MAIN             99D, 116D
             smc1.f/abc_c1_invert     297D, 314D, 375, 389
              " /sys_matrix_c1        32D, 50D, 109, 111, 133, 157
vxp_cy1_b2   c1.f/abc_rc1_invert      552D, 569D
              " /c1a_pot              33D, 51D
              " /c1b_pot              119D, 137D, 172=
              " /output_rc1           667D, 684D
              " /sys_matrix_rc1       210D, 228D, 284, 307, 335, 360
             c2.f/abc_rc2_invert      819D, 836D
              " /output_rc2           1009D, 1026D
              " /sys_matrix_rc2       217D, 235D, 476, 509, 551, 583
             mr2cf.f/MAIN             100D, 117D
             smc1.f/abc_c1_invert     298D, 315D, 376, 390
              " /sys_matrix_c1        33D, 51D, 113, 115, 135, 160
vxp_cy2_b1   c2.f/abc_rc2_invert      854D, 873D
              " /c2b_pot              33D, 51D, 86=
              " /c2c_pot              124D, 143D
              " /output_rc2           1044D, 1063D, 1158, 1239, 1279, 1302,
                             1346, 1372, 1390
              " /sys_matrix_rc2       255D, 274D, 457, 493, 530, 569
             mr2cf.f/MAIN             137D, 155D
vxp_cy2_b2   c2.f/abc_rc2_invert      855D, 874D
              " /c2b_pot              34D, 52D, 89=
              " /c2c_pot              125D, 144D
              " /output_rc2           1045D, 1064D, 1161, 1241, 1281, 1304,
                             1348, 1374, 1392
              " /sys_matrix_rc2       256D, 275D, 460, 495, 533, 571
             mr2cf.f/MAIN             138D, 156D
```

Variables :

```
Symbol     File/Subprogram        Line
------     ---- ----------        ----

vxp_cy2_c1  c2.f/abc_rc2_invert    856D, 875D, 954, 975
             " /c2b_pot            35D, 53D
             " /c2c_pot            126D, 145D, 179=
             " /output_rc2         1046D, 1065D
             " /sys_matrix_rc2     257D, 276D, 351, 353, 387, 422
            mr2cf.f/MAIN           139D, 157D
vxp_cy2_c2  c2.f/abc_rc2_invert    857D, 876D, 955, 976
             " /c2b_pot            36D, 54D
             " /c2c_pot            127D, 146D, 182=
             " /output_rc2         1047D, 1066D
             " /sys_matrix_rc2     258D, 277D, 355, 357, 389, 425
            mr2cf.f/MAIN           140D, 158D
vxp_rod     c1.f/abc_rc1_invert    581D, 587D
             " /output_rc1         696D, 702D
             " /sys_matrix_rc1     240D, 246D, 396, 420, 450, 473
            c2.f/abc_rc2_invert    890D, 896D
             " /output_rc2         1079D, 1085D
             " /sys_matrix_rc2     291D, 297D
            mr2cf.f/MAIN           68D, 72D
            rf.f/abc_rod_invert    209D, 215D
             " /abc_rod_solve      159D, 165D
             " /output             272D, 278D, 315, 348, 367, 378, 401,
                                   416, 424
             " /rod_pot            23D, 28D, 50=
             " /sys_matrix_rod     72D, 78D, 106, 113, 122, 139
wc          c1.f/output_rc1        720D, 809=, 811
            c2.f/output_rc2        1115D, 1208=, 1210
work        mr2cf.f/MAIN           35D
            rf.f/abc_rod_invert    232D, 237=
             " /minv               448D/*, 456D, 465=, 471=, 472=, 479,
                                   482, 485, 490, 491, 491=, 492=, 502,
                                   502, 502, 503, 503=, 505=, 513, 515,
                                   515=, 521, 523, 523, 523=, 530
workc1      mr2cf.f/MAIN           59D
            smc1.f/abc_c1_invert   351D, 357=
workrc1     c1.f/abc_rc1_invert    604D, 610=
            mr2cf.f/MAIN           48D
workrc2     c2.f/abc_rc2_invert    925D, 931=
            mr2cf.f/MAIN           54D
wr          rf.f/output            286D, 336=, 338
```

Variables :

Symbol	File/Subprogram	Line
------	---- ----------	----
y2	cbessl.f/d1cbessy	362D, 371=, 372, 376=, 379, 388=, 389
	" /d2cbessy	407D, 413=, 414, 418=, 421, 425=, 430, 440=, 442
y3	" /d1cbessy	362D, 377=, 379
	" /d2cbessy	407D, 419=, 421, 426=, 430
y4	" / "	407D, 428=
yn	mr2cf.f/MAIN	8D
z	cbessl.f/cbessi	145D, 148=, 155, 158
	" /cbessj	62D, 70=, 72, 76, 77, 88, 91, 126, 126, 127
	" /cbessk	248D, 253=, 254, 267, 271, 279, 283
	" /cbessy	171D, 177=, 178, 180, 195, 199, 207, 211, 225, 227, 229, 230, 232, 232, 233
	mr2cf.f/MAIN	8D
z2	cbessl.f/cbessk	248D, 254=, 273
	" /cbessy	171D, 178=, 201
zero	c1.f/output_rc1	654D, 727=, 868, 869, 891, 892, 914, 915, 940, 941, 958, 959, 984, 985, 1002, 1003
	c2.f/output_rc2	996D, 1123=, 1266, 1267, 1289, 1290, 1312, 1313, 1338, 1339, 1356, 1357, 1382, 1383, 1400, 1401
	rf.f/output	269D, 292=, 358, 359, 370, 371, 382, 383, 396, 397, 405, 406, 419, 420, 428, 429
zeta_1cyl	mr2cf.f/MAIN	25D, 200=, 209
zeta_2cyl	" / "	26D, 223=, 232
zeta_rod	" / "	21D, 185=, 189
zm	cbessl.f/cbessj	61D, 72=, 79
	" /cbessy	170D, 180=, 182

LABELS

Labels :

Symbol	File/Subprogram	Line
------	---- ----------	----
4	mr2cf.f/MAIN	488, 489D
10	cbessl.f/cbessi	153, 157D
	" /cbessj	86, 90D
	" /cbessk	265, 269D
	" /cbessy	193, 197D
	" /fac	27, 29D
	" /gamma	10, 12D
	" /psi	47, 49D
	mr2cf.f/MAIN	299, 300D
20	cbessl.f/cbessk	275, 281D
	" /cbessy	203, 209D
	mr2cf.f/MAIN	294, 295D
51	" / "	476, 477D
60	" / "	478, 479D
70	" / "	480, 481D
100	rf.f/minv	464, 466D
110	" / "	463, 467D
120	" / "	470, 473D
130	" / "	469, 474D
140	" / "	478D, 510
150	" / "	481, 483, 486D
160	" / "	489, 493D
170	" / "	488, 495D
180	" / "	501, 504D
190	" / "	500, 506D
200	" / "	514, 516D
210	" / "	512, 517D
220	" / "	522, 524D
230	" / "	520, 525D
240	" / "	519, 526D
250	" / "	529, 531D
260	" / "	528, 532D
400	mr2cf.f/MAIN	334, 335D, 367, 404, 435, 458
410	" / "	427, 428D
900	rf.f/minv	499, 552D

STRINGS

Strings :

```
Symbol      File/Subprogram       Line
------      ---- ----------       ----

'  r = '  mr2cf.f/MAIN            300
' D_IF = '    "  /  "            428
' IFSC = '    "  /  "            428
' M_OF = '    "  /  "            335
' OFSC = '    "  /  "            335
' Om = '      "  /  "            482
' ao_1cyl = '
              "  /  "            301
' bo_c1 = '   "  /  "            301
' co_c2 = '   "  /  "            301
'BMAG < ARTIFICIAL ZERO (1.0D-12) RETURNING FROM MINV'
          rf.f/minv              555
'Enter tft,k,f,emt: (where emt=1 1CYLINDER, emt=2 ROD, emt=3 ROD&CYL, emt=4
2CYLINDERS,             using mr2cf.f)'
          mr2cf.f/MAIN           290
'MAIN PROGRAM  n = '
              "  /  "            300
'SUB OUTPUT               k = '
              "  /  "            481
'SUB OUTPUT THE ANSWER IS g(jk) = '
              "  /  "            479
'SUB OUTPUT THE ANSWER IS value = '
              "  /  "            477
'Type 1 if you wish to quit'
              "  /  "            487
'd1_IFSC = '  "  /  "            428
'd1_OFSC = '  "  /  "            335
'tft = '      "  /  "            302
```

INDEX

Index Thu Oct 19 16:48:44 1995

General options in effect:
-o mr2cf.lst

Files:

Cross Reference:

Legend:

D	Definition/declaration
=	Possibly modified occurrence
*	Dummy argument

APPENDIX A

INTERMEDIATE VARIABLES FOR NUWC-NPT TR 11,043

Table A-1. Common Block /ROD/

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Solid Cylinder Variables)
SP_rod	$J_n(p_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_1 r_1)$.
d1_SP__rod	$\frac{\partial}{\partial r} J_n(p_1 a)$	First derivative of $J_n(p_1 r)$ with respect to r. Can be evaluated again for $(p_1 r_1)$.
d2_SP_rod	$\frac{\partial^2}{\partial r^2} J_n(p_1 a)$	Second derivative of $J_n(p_1 r)$ with respect to r. Can be evaluated again for $(p_1 r_1)$.
VXP_rod	$J_n(q_1 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_1 r_1)$.
d1_VXP_rod	$\frac{\partial}{\partial r} J_n(q_1 a)$	First derivative of $J_n(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
d2_VXP_rod	$\frac{\partial^2}{\partial r^2} J_n(q_1 a)$	Second derivative of $J_n(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
VRTP_rod	$J_{n+1}(q_1 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_1 r_1)$.
d1_VRTP_rod	$\frac{\partial}{\partial r} J_{n+1}(q_1 a)$	First derivative of $J_{n+1}(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
d2_VRTP_rod	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_1 a)$	Second derivative of $J_{n+1}(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.

Table A-1. Common Block /ROD/ (Cont'd)

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Solid Cylinder Variables)
A1_rod	A_1^{C1}	Undetermined constant.
B1_rod	B_1^{C1}	Undetermined constant.
C1_rod	C_1^{C1}	Undetermined constant.
lame_rod	λ_1	Lame constant for the solid cylinder.
shear_rod	μ_1	Shear modulus for the solid cylinder.

Table A-2. Common Block /CYLINDER1/

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
SP_CY1_a1	$J_n(p_2a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for (p_2r_1).
d1_SP_CY1_a1	$\frac{\partial}{\partial r}J_n(p_2a)$	First derivative of $J_n(p_2r)$ with respect to r. Can be evaluated again for (p_2r_1).
d2_SP_CY1_a1	$\frac{\partial^2}{\partial r^2}J_n(p_2a)$	Second derivative of $J_n(p_2r)$ with respect to r. Can be evaluated again for (p_2r_1).
VXP_CY1_a1	$J_n(q_2a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for (q_2r_1).
d1_VXP_CY1_a1	$\frac{\partial}{\partial r}J_n(q_2a)$	First derivative of $J_n(q_2r)$ with respect to r. Can be evaluated again for (q_2r_1).
d2_VXP_CY1_a1	$\frac{\partial^2}{\partial r^2}J_n(q_2a)$	Second derivative of $J_n(q_2r)$ with respect to r. Can be evaluated again for (q_2r_1).

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
VRTP_CY1_a1	$J_{n+1}(q_2 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_2 r_1)$.
d1_VRTP_CY1_a1	$\frac{\partial}{\partial r} J_{n+1}(q_2 a)$	First derivative of $J_{n+1}(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
d2_VRTP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_2 a)$	Second derivative of $J_{n+1}(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
SP_CY1_b1	$J_n(p_2 b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b1	$\frac{\partial}{\partial r} J_n(p_2 b)$	First derivative of $J_n(p_2 r)$ with respect to r.
d2_SP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(p_2 b)$	Second derivative of $J_n(p_2 r)$ with respect to r.
VXP_CY1_b1	$J_n(q_2 b)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY1_b1	$\frac{\partial}{\partial r} J_n(q_2 b)$	First derivative of $J_n(q_2 r)$ with respect to r.
d2_VXP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(q_2 b)$	Second derivative of $J_n(q_2 r)$ with respect to r.
VRTP_CY1_b1	$J_{n+1}(q_2 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
d1_VRTP_CY1_b1	$\frac{\partial}{\partial r}J_{n+1}(q_2b)$	First derivative of $J_{n+1}(q_2r)$ with respect to r.
d2_VRTP_CY1_b1	$\frac{\partial^2}{\partial r^2}J_{n+1}(q_2b)$	Second derivative of $J_{n+1}(q_2r)$ with respect to r
SP_CY1_a2	$Y_n(p_2a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again at (p_2r_1).
d1_SP_CY1_a2	$\frac{\partial}{\partial r}Y_n(p_2a)$	First derivative of $Y_n(p_2r)$ with respect to r. Can be evaluated again at (p_2r_1).
d2_SP_CY1_a2	$\frac{\partial^2}{\partial r^2}Y_n(p_2a)$	Second derivative of $Y_n(p_2r)$ with respect to r. Can be evaluated again at (p_2r_1).
VXP_CY1_a2	$Y_n(q_2a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again at (q_2r_1).
d1_VXP_CY1_a2	$\frac{\partial}{\partial r}Y_n(q_2a)$	First derivative of $Y_n(q_2r)$ with respect to r. Can be evaluated again at (q_2r_1).
d2_VXP_CY1_a2	$\frac{\partial^2}{\partial r^2}Y_n(q_2a)$	Second derivative of $Y_n(q_2r)$ with respect to r. Can be evaluated again at (q_2r_1).
VRTP_CY1_a2	$Y_{n+1}(q_2a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again at (q_2r_1).
d1_VRTP_CY1_a2	$\frac{\partial}{\partial r}Y_{n+1}(q_2a)$	First derivative of $Y_{n+1}(q_2r)$ with respect to r. Can be evaluated again at (q_2r_1).

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
d2_VRTP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2 a)$	Second derivative of $Y_{n+1}(q_2 r)$ with respect to r. Can be evaluated again at $(q_2 r_1)$.
SP_CY1_b2	$Y_n(p_2 b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b2	$\frac{\partial}{\partial r} Y_n(p_2 b)$	First derivative of $Y_n(p_2 r)$ with respect to r.
d2_SP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 b)$	Second derivative of $Y_n(p_2 r)$ with respect to r.
VXP_CY1_b2	$Y_n(q_2 b)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY1_b2	$\frac{\partial}{\partial r} Y_n(q_2 b)$	First derivative of $Y_n(q_2 r)$ with respect to r.
d2_VXP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 b)$	Second derivative of $Y_n(q_2 r)$ with respect to r.
VRTP_CY1_b2	$Y_{n+1}(q_2 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY1_b2	$\frac{\partial}{\partial r} Y_{n+1}(q_2 b)$	First derivative of $Y_{n+1}(q_2 r)$ with respect to r.
d2_VRTP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2 b)$	Second derivative of $Y_{n+1}(q_2 r)$ with respect to r.
A1_C1	A_1^{C2}	Undetermined constant.
A2_C1	A_2^{C2}	Undetermined constant.

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
B1_C1	B_1^{C2}	Undetermined constant.
B2_C1	B_2^{C2}	Undetermined constant.
C1_C1	C_1^{C2}	Undetermined constant.
C2_C1	C_2^{C2}	Undetermined constant.
lame_c1	λ_2	Lame constant for the second cylinder.
shear_c1	μ_2	Shear modulus for the second cylinder.

Table A-3. Calculated Solid Cylinder Variable

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
Ec_rod	E_1^*	Complex Young's modulus.

Table A-4. Calculated Outer Cylinder Variables

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
Ec_c1	E_2^*	Complex Young's modulus.
bo_c1	b	Outer radius.

Table A-5. Common Block /OFLUID/

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
OFSC	$H_n^{(1)}(g_2 b)$ or $K_n(f_2 b)$	Modified Bessel function, used for the scalar displacement potential. Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
d1_OFSC	$\frac{\partial}{\partial r} H_n^{(1)}(g_2 b)$ or $\frac{\partial}{\partial r} K_n(f_2 b)$	First derivative of "OFSC" with respect to r. Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
M_OF	M or H	Undetermined constant.

Table A-6. System Matrix Variables

FORTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
sm	SR	System matrix for the single solid cylinder and outer fluid.
smrc1	S	System matrix for the double cylinder and outer fluid.

APPENDIX B

INTERMEDIATE VARIABLES FOR NUWC-NPT TR 11,067

Table B-1. Common Block /CYLINDER1/

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
SP_CY1_a1	$J_n(p_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_1 r_1)$.
d1_SP_CY1_a1	$\frac{\partial}{\partial r} J_n(p_1 a)$	First derivative of $J_n(p_1 r)$ with respect to r. Can be evaluated again for $(p_1 r_1)$.
d2_SP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_n(p_1 a)$	Second derivative of $J_n(p_1 r)$ with respect to r. Can be evaluated again for $(p_1 r_1)$.
VXP_CY1_a1	$J_n(q_1 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_1 r_1)$.
d1_VXP_CY1_a1	$\frac{\partial}{\partial r} J_n(q_1 a)$	First derivative of $J_n(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
d2_VXP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_n(q_1 a)$	Second derivative of $J_n(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
VRTP_CY1_a1	$J_{n+1}(q_1 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_1 r_1)$.
d1_VRTP_CY1_a1	$\frac{\partial}{\partial r} J_{n+1}(q_1 a)$	First derivative of $J_{n+1}(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
d2_VRTP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_1 a)$	Second derivative of $J_{n+1}(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.

Table B-1. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
SP_CY1_b1	$J_n(p_1 b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b1	$\frac{\partial}{\partial r} J_n(p_1 b)$	First derivative of $J_n(p_1 r)$ with respect to r.
d2_SP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(p_1 b)$	Second derivative of $J_n(p_1 r)$ with respect to r.
VXP_CY1_b1	$J_n(q_1 b)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY1_b1	$\frac{\partial}{\partial r} J_n(q_1 b)$	First derivative of $J_n(q_1 r)$ with respect to r.
d2_VXP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(q_1 b)$	Second derivative of $J_n(q_1 r)$ with respect to r.
VRTP_CY1_b1	$J_{n+1}(q_1 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY1_b1	$\frac{\partial}{\partial r} J_{n+1}(q_1 b)$	First derivative of $J_{n+1}(q_1 r)$ with respect to r.
d2_VRTP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_1 b)$	Second derivative of $J_{n+1}(q_1 r)$ with respect to r.
SP_CY1_a2	$Y_n(p_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_1 r_1)$.
d1_SP_CY1_a2	$\frac{\partial}{\partial r} Y_n(p_1 a)$	First derivative of $Y_n(p_1 r)$ with respect to r. Can be evaluated again for $(p_1 r_1)$.

Table B-1. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d2_SP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_n(p_1 a)$	Second derivative of $Y_n(p_1 r)$ with respect to r. Can be evaluated again for $(p_1 r_1)$.
VXP_CY1_a2	$Y_n(q_1 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_1 r_1)$.
d1_VXP_CY1_a2	$\frac{\partial}{\partial r} Y_n(q_1 a)$	First derivative of $Y_n(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
d2_VXP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_n(q_1 a)$	Second derivative of $Y_n(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
VRTP_CY1_a2	$Y_{n+1}(q_1 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_1 r_1)$.
d1_VRTP_CY1_a2	$\frac{\partial}{\partial r} Y_{n+1}(q_1 a)$	First derivative of $Y_{n+1}(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
d2_VRTP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_1 a)$	Second derivative of $Y_{n+1}(q_1 r)$ with respect to r. Can be evaluated again for $(q_1 r_1)$.
SP_CY1_b2	$Y_n(p_1 b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b2	$\frac{\partial}{\partial r} Y_n(p_1 b)$	First derivative of $Y_n(p_1 r)$ with respect to r.
d2_SP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(p_1 b)$	Second derivative of $Y_n(p_1 r)$ with respect to r.
VXP_CY1_b2	$Y_n(q_1 b)$	Bessel function of the first kind, used for the vector x displacement potential.

Table B-1. Common Block /CYLINDER1/ (Cont'd)

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d1_VXP_CY1_b2	$\frac{\partial}{\partial r}Y_n(q_1 b)$	First derivative of $Y_n(q_1 r)$ with respect to r.
d2_VXP_CY1_b2	$\frac{\partial^2}{\partial r^2}Y_n(q_1 b)$	Second derivative of $Y_n(q_1 r)$ with respect to r.
VRTP_CY1_b2	$Y_{n+1}(q_1 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY1_b2	$\frac{\partial}{\partial r}Y_{n+1}(q_1 b)$	First derivative of $Y_{n+1}(q_1 r)$ with respect to r.
d2_VRTP_CY1_b2	$\frac{\partial^2}{\partial r^2}Y_{n+1}(q_1 b)$	Second derivative of $Y_{n+1}(q_1 r)$ with respect to r.
A1_C1	A_1^{C1}	Undetermined constant.
A2_C1	A_2^{C1}	Undetermined constant.
B1_C1	B_1^{C1}	Undetermined constant.
B2_C1	B_2^{C1}	Undetermined constant.
C1_C1	C_1^{C1}	Undetermined constant.
C2_C1	C_2^{C1}	Undetermined constant.
lame_c1	λ_1	Lame constant for cylinder 1.
shear_c1	μ_1	Shear modulus for cylinder 1.

Table B-2. Common Block /CYLINDER2/

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
SP_CY2_b1	$J_n(p_2 b)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_2 r_1)$.
d1_SP_CY2_b1	$\frac{\partial}{\partial r}J_n(p_2 b)$	First derivative of $J_n(p_2 r)$ with respect to r. Can be evaluated again for $(p_2 r_1)$.
d2_SP_CY2_b1	$\frac{\partial^2}{\partial r^2}J_n(p_2 b)$	Second derivative of $J_n(p_2 r)$ with respect to r. Can be evaluated again for $(p_2 r_1)$.
VXP_CY2_b1	$J_n(q_2 b)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_2 r_1)$.
d1_VXP_CY2_b1	$\frac{\partial}{\partial r}J_n(q_2 b)$	First derivative of $J_n(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
d2_VXP_CY2_b1	$\frac{\partial^2}{\partial r^2}J_n(q_2 b)$	Second derivative of $J_n(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
VRTP_CY2_b1	$J_{n+1}(q_2 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_2 r_1)$.
d1_VRTP_CY2_b1	$\frac{\partial}{\partial r}J_{n+1}(q_2 b)$	First derivative of $J_{n+1}(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
d2_VRTP_CY2_b1	$\frac{\partial^2}{\partial r^2}J_{n+1}(q_2 b)$	Second derivative of $J_{n+1}(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
SP_CY2_c1	$J_n(p_2 c)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY2_c1	$\frac{\partial}{\partial r}J_n(p_2 c)$	First derivative of $J_n(p_2 r)$ with respect to r.

Table B-2. Common Block /CYLINDER2/ (Cont'd)

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d2_SP_CY2_c1	$\frac{\partial^2}{\partial r^2} J_n(p_2 c)$	Second derivative of $J_n(p_2 r)$ with respect to r.
VXP_CY2_c1	$J_n(q_2 c)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY2_c1	$\frac{\partial}{\partial r} J_n(q_2 c)$	First derivative of $J_n(q_2 r)$ with respect to r.
d2_VXP_CY2_c1	$\frac{\partial^2}{\partial r^2} J_n(q_2 c)$	Second derivative of $J_n(q_2 r)$ with respect to r.
VRTP_CY2_c1	$J_{n+1}(q_2 c)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY2_c1	$\frac{\partial}{\partial r} J_{n+1}(q_2 c)$	First derivative of $J_{n+1}(q_2 r)$ with respect to r.
d2_VRTP_CY2_c1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_2 c)$	Second derivative of $J_{n+1}(q_2 r)$ with respect to r.
SP_CY2_b2	$Y_n(p_2 b)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_2 r_1)$.
d1_SP_CY2_b2	$\frac{\partial}{\partial r} Y_n(p_2 b)$	First derivative of $Y_n(p_2 r)$ with respect to r. Can be evaluated again for $(p_2 r_1)$.
d2_SP_CY2_b2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 b)$	Second derivative of $Y_n(p_2 r)$ with respect to r. Can be evaluated again for $(p_2 r_1)$.
VXP_CY2_b2	$Y_n(q_2 b)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_2 r_1)$.

Table B-2. Common Block /CYLINDER2/ (Cont'd)

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d1_VXP_CY2_b2	$\frac{\partial}{\partial r} Y_n(q_2 b)$	First derivative of $Y_n(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
d2_VXP_CY2_b2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 b)$	Second derivative of $Y_n(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
VRTP_CY2_b2	$Y_{n+1}(q_2 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_2 r_1)$.
d1_VRTP_CY2_b2	$\frac{\partial}{\partial r} Y_{n+1}(q_2 b)$	First derivative of $Y_{n+1}(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
d2_VRTP_CY2_b2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2 b)$	Second derivative of $Y_{n+1}(q_2 r)$ with respect to r. Can be evaluated again for $(q_2 r_1)$.
SP_CY2_c2	$Y_n(p_2 c)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY2_c2	$\frac{\partial}{\partial r} Y_n(p_2 c)$	First derivative of $Y_n(p_2 r)$ with respect to r.
d2_SP_CY2_c2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 c)$	Second derivative of $Y_n(p_2 r)$ with respect to r.
VXP_CY2_c2	$Y_n(q_2 c)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY2_c2	$\frac{\partial}{\partial r} Y_n(q_2 c)$	First derivative of $Y_n(q_2 r)$ with respect to r.
d2_VXP_CY2_c2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 c)$	Second derivative of $Y_n(q_2 r)$ with respect to r.

Table B-2. Common Block /CYLINDER2/ (Cont'd)

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
VRTP_CY2_c2	$Y_{n+1}(q_2c)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY2_c2	$\frac{\partial}{\partial r}Y_{n+1}(q_2c)$	First derivative of $Y_{n+1}(q_2r)$ with respect to r.
d2_VRTP_CY2_c2	$\frac{\partial^2}{\partial r^2}Y_{n+1}(q_2c)$	Second derivative of $Y_{n+1}(q_2r)$ with respect to r.
A1_C2	A_1^{C2}	Undetermined constant.
A2_C2	A_2^{C2}	Undetermined constant.
B1_C2	B_1^{C2}	Undetermined constant.
B2_C2	B_2^{C2}	Undetermined constant.
C1_C2	C_1^{C2}	Undetermined constant.
C2_C2	C_2^{C2}	Undetermined constant.
lame_c2	λ_2	Lame constant for cylinder 2.
shear_c2	μ_2	Shear modulus for cylinder 2.

Table B-3. Common Block /IFLUID/

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
IFSC	$J_n(g_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can also be evaluated at $(g_1 r_1)$.
d1_IFSC	$\frac{\partial}{\partial r} J_n(g_1 a)$	First derivative of $J_n(g_1 r)$ with respect to r. Can also be evaluated at $(g_1 r_1)$.
D_IF	D	Undetermined constant.

Table B-4. Common Block /OFLUID/

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
OFSC	$H_n^{(1)}(g_2 b)$ or $K_n(f_2 b)$	Modified Bessel function, used for the scalar displacement potential. Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
d1_OFSC	$\frac{\partial}{\partial r} H_n^{(1)}(g_2 b)$ or $\frac{\partial}{\partial r} K_n(f_2 b)$	First derivative of "OFSC" with respect to r. Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
M_OF	M or H	Undetermined constant.

Table B-5. Calculated Cylinder 1 Variables

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
Ec_c1	E_1^*	Complex Young's modulus.
bo_c1	b	Outer radius.

Table B-6. Calculated Cylinder 2 Variables

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
Ec_c2	E_2^*	Complex Young's modulus.
co_c2	c	Outer radius.

Table B-7. System Matrix Variables

FORTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
smrc1	$smc1$	System matrix for the single cylinder with inner and outer fluids.
smrc2	C	System matrix for the double cylinder with inner and outer fluids.

APPENDIX C

MAKEFILE

```
# Makefile for mr2cf program
# PATH /home/kwcode/analysis/tota/RODCY2
#
#
all:mr2cf
#
mr2cf:mr2cf.o zbessel.o cbessl.o rf.o c1.o c2.o fluids.o smc1.o
    f77 -g -u mr2cf.o zbessel.o cbessl.o rf.o c1.o c2.o fluids.o smc1.o -Nl20 -lm -o mr2cf
#
zbessel:zbessel.o
    f77 -g -u zbessel.f
#
#
cbessl: cbessl.o
    f77 -g -u cbessl.f
#
#
rf:  rf.o
    f77 -g -u rf.f
#
#
c1: c1.o
    f77 -g -u c1.f
#
c2:    c2.o
    f77 -g -u c2.f
#
fluids:    fluids.o
    f77 -g -u fluids.f

smc1:smc1.o
    f77 -g -u smc1.f
#
clean:
    rm -f *.o *%
#
```

APPENDIX D

SUBPROUTINE cbessl.f_matlab

```
C ***************************************************************************
C
C               subroutine "cbessl.f_matlab"
C
C This subroutine was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C ***************************************************************************
C This subroutine "cbessel.f_matlab" uses the Matlab library for Bessel Functions
C of complex argument z. The source file needed is zbessel.f. The following
C Bessel functions are used from Matlab: J, Y, and K.
C The formula for the first derivative of the Bessel functions is taken from
C Conduction Heat Transfer by Vedat S. Arpaci. They are also listed in
C Field Theory Handbook by Moon and Spencer.
C The formula for the second derivative was derived by Mark S. Peloquin and checked
C by Roy Streit. It is listed as B-9 in the notes.
C This code was written by Mark S. Peloquin 6/10/95 at NUWCDETNLON.


      function gamma(n)
      integer i,n,sum,gamma
      if(n .eq. 0 .or. n .eq. 1) then
         gamma = 1
      else
         sum=n
         do 10, i=1, n-1
            sum = sum*(n-i)
 10      continue
         gamma = sum
      endif
      return
      end
```

```
      function fac(n)
       integer n,sum,fac,i
       if(n .eq. 0 .or. n .eq. 1) then
            fac = 1
       else
            sum=n
            do 10, i=1, n-1
                 sum = sum*(n-i)
10           continue
            fac = sum
       endif
       return
       end
```

```
      function psi(n)
      integer n,na,i
      real*8 sum,psi,euler
      parameter (euler=.5772156649015328606)
      sum = 0.0
      na = iabs(n)
      if (na .eq. 1) then
       psi = - euler
      else
       do 10, i=1, na-1
        sum = sum + (1.0/i)
10     continue
       psi = -euler + sum
      endif
      return
      end
```

```
      double complex function cbessj(n,a,r)

      complex*16 a
      real*8 r
      integer n

      COMPLEX*16 Z,W,W1,W2,WRK
      REAL*8 ZR,ZI,NU,PI,D1MACH
      INTEGER KODE,UNFL,IERR
      DATA PI /3.14159265358979323850D0/
C
C  Pick up input arguments and allocate output array
C

      nu = n
      Z = a*r
      KODE = 1
C
C     J_nu(z)
C
        IF (NU .GE. 0.0D0) THEN
          CALL CBESJ(Z,NU,KODE,1,W,UNFL,IERR)
        ELSE IF (NU .EQ. DINT(NU)) THEN
          CALL CBESJ(Z,-NU,KODE,1,W,UNFL,IERR)
          IF (MOD(IDINT(NU),2) .NE. 0) W = -W
        ELSE IF (ZR .EQ. 0.0D0 .AND. ZI .EQ. 0.0D0) THEN
          W = -D1MACH(6)
        ELSE
          CALL CBESJ(Z,-NU,KODE,1,W1,UNFL,IERR)
          CALL CBESY(Z,-NU,KODE,1,W2,UNFL,WRK,IERR)
          W = DCOS(NU*PI)*W1 + SIN(NU*PI)*W2
        ENDIF

       cbessj = W

        return
        end
```

```
      double complex function cbessy(n,a,r)

      complex*16 a
      real*8 r
      integer n

      COMPLEX*16 Z,W,W1,W2,WRK
      REAL*8 ZR,ZI,NU,PI,D1MACH
      INTEGER KODE,UNFL,IERR
      DATA PI /3.14159265358979323850D0/
C
C  Pick up input arguments and allocate output array
C

      NU   = n
      Z    = a*r
      KODE = 1
      ZR   = DREAL(Z)
      ZI   = DIMAG(Z)

C
C       Y_nu(z)
C       Y(nu,0) = -Infinity
C
        IF (ZR .EQ. 0.0D0 .AND. ZI .EQ. 0.0D0) THEN
          W = -D1MACH(6)
        ELSE IF (NU .GE. 0.0D0) THEN
          CALL CBESY(Z,NU,KODE,1,W,UNFL,WRK,IERR)
        ELSE IF (NU .EQ. DINT(NU)) THEN
          CALL CBESY(Z,-NU,KODE,1,W,UNFL,WRK,IERR)
          IF (MOD(IDINT(NU),2) .NE. 0) W = -W
        ELSE
          CALL CBESJ(Z,-NU,KODE,1,W1,UNFL,IERR)
          CALL CBESY(Z,-NU,KODE,1,W2,UNFL,WRK,IERR)
          W = DCOS(NU*PI)*W2 - SIN(NU*PI)*W1
        ENDIF
C

      cbessy = W


      return
      end
```

```
C  Argument limit is approximately (3.0,3.0) for cbessi
     double complex function cbessi(n,a,r)

     integer n,limit,j,k,na,fac
     real*8 r
     complex*16 a,z,sum,total
    double complex cbessi

     z = a*r
    na=iabs(n)
     limit = 10
     total = (0.0,0.0)

     do 10, k=0, limit
      j = na + k
      sum = ((.25*z**2)**k)/(fac(k)*fac(j))
      total = total + sum
10    continue
     cbessi = ((z/2)**na)*total


     return
     end


    double complex function cbessk(n,a,r)

    complex*16 a
    real*8 r
    integer n

    COMPLEX*16 Z,W
    REAL*8 ZR,ZI,NU,PI,D1MACH
    INTEGER FUN,KODE,UNFL,IERR
    DATA PI /3.14159265358979323 85D0/
C
C   Pick up input arguments and allocate output array
C

    NU   = n
    Z    = a*r
    KODE = 1
    ZR   = DREAL(Z)
    ZI   = DIMAG(Z)
    FUN  = 75
```

```
C
C       K_nu(z)
C       K(nu,0) = Infinity
C
      IF (FUN .EQ. 75) THEN
        IF (ZR .EQ. 0.0D0 .AND. ZI .EQ. 0.0D0) THEN
          W = D1MACH(6)
        ELSE
          CALL CBESK(Z,DABS(NU),KODE,1,W,UNFL,IERR)
        ENDIF
      ENDIF
C
    cbessk = W

    return
    end



    double complex function d1cbessk(n,a,r)

    real*8 r
    complex*16 a
    integer n
     double complex cbessk,d1cbessk

    d1cbessk = -a*cbessk(n+1,a,r)+(n/r)*cbessk(n,a,r)

    return
    end



    double complex function d2cbessk(n,a,r)

     real*8 r
     complex*16 a
     integer n
     double complex d2cbessk

C Need the proper equation here, not being used 6/10/95.

    d2cbessk = (1.0,1.0)
```

```
      return
      end



      double complex function d1cbessj(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessj,d1cbessj

      d1cbessj = -a*cbessj(n+1,a,r)+(n/r)*cbessj(n,a,r)

      return
      end



      double complex function d2cbessj(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessj,d2cbessj


      d2cbessj = ((a**2)/4.0D0)*(cbessj(n-2,a,r) - 2.0D0*
     1     cbessj(n,a,r) + cbessj(n+2,a,r))

      return
      end
```

```
      double complex function d1cbessy(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessy,d1cbessy

      d1cbessy = -a*cbessy(n+1,a,r)+(n/r)*cbessy(n,a,r)

      return
      end



      double complex function d2cbessy(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessy,d2cbessy


      d2cbessy = ((a**2)/4.0D0)*(cbessy(n-2,a,r) - 2.0D0*
     1     cbessy(n,a,r) + cbessy(n+2,a,r))


      return
      end




      double complex function d1cbessi(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex cbessi,d1cbessi

      d1cbessi = a*cbessi(n+1,a,r)+(n/r)*cbessi(n,a,r)

      return
      end
```

```
      double complex function d2cbessi(n,a,r)

      real*8 r
      complex*16 a
      integer n
      double complex d2cbessi

C Need the proper equation here, not being used 6/10/95.

      d2cbessi = (1.0,1.0)

      return
      end



     double complex function cbessh1(n,a,r)
      real*8 r
      complex*16 a
      integer n
     double complex cbessj,cbessy,cbessh1

     cbessh1 = cbessj(n,a,r)+(0.0, 1.0)*cbessy(n,a,r)

     return
     end



      double complex function cbessh2(n,a,r)
      real*8 r
      integer n
      complex*16 a
      double complex cbessj,cbessy,cbessh2

      cbessh2 = cbessj(n,a,r)-(0.0, 1.0)*cbessy(n,a,r)

      return
      end
```

```
      double complex function d1cbessh1(n,a,r)
      integer n
      real*8 r
      complex a
      double complex d1cbessj,d1cbessy,d1cbessh1

      d1cbessh1 = d1cbessj(n,a,r) + (0.0,1.0)*d1cbessy
     1(n,a,r)

      return
      end
```

```
      double complex function d1cbessh2(n,a,r)
      integer n
      real*8 r
      complex a
      double complex d1cbessj,d1cbessy,d1cbessh2

      d1cbessh2 = d1cbessj(n,a,r) - (0.0,1.0)*d1cbessy
     1(n,a,r)

      return
      end
```

APPENDIX E

MODIFIED SUBROUTINES FOR CIRCUMFERENTIAL EXCITATION

```
      SUBROUTINE ABC_ROD_INVERT(exctype,sm)

C EXTERNAL VARIABLES

      integer exctype
      complex*16 sm(4,4)

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      complex*16 OFSC,d1_OFSC,M_OF

      common /OFLUID/ OFSC,d1_OFSC,M_OF

C ****************************************************************

C INTERNAL VARIABLES

      integer n,iflag
      complex*16 sminv(4,4),work(4,8)

      n = 4
      iflag = 0

      CALL MINV(sm,sminv,work,n,iflag)
```

```
C CIRCUMFERENTIAL      EXCITATION   exctype = 2
C RADIAL               EXCITATION   exctype = 1
C AXIAL                EXCITATION   exctype = 0

      if (exctype .eq. 1) then
        A1_rod = -sminv(1,1)
        B1_rod = -sminv(2,1)
        C1_rod = -sminv(3,1)
         M_OF = -sminv(4,1)
      elseif (exctype .eq. 0) then
        A1_rod = -sminv(1,2)
        B1_rod = -sminv(2,2)
        C1_rod = -sminv(3,2)
         M_OF = -sminv(4,2)
      elseif (exctype .eq. 2) then
        A1_rod = -sminv(1,3)
        B1_rod = -sminv(2,3)
        C1_rod = -sminv(3,3)
         M_OF = -sminv(4,3)
      endif


      return
      end
```

```
SUBROUTINE ABC_C1_INVERT(n,exctype,smc1,a,b)

C EXTERNAL VARIABLES

      integer n,exctype
      real*8 a,b
      complex*16 smc1(7,7)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
    1          SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
    1          SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
    1          SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
    1          VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
    1          VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
    1          VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
    1          VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
    1          VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
    1          VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
    1          VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
    1          VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
    1          lame_c1,shear_c1,cl_c1,ct_c1,
    1          A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C ****************************************************************
```

```
C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      complex*16 OFSC,d1_OFSC,M_OF

      common /OFLUID/ OFSC,d1_OFSC,M_OF

C *****************************************************************


C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

      common /IFLUID/ IFSC,d1_IFSC,D_IF

      complex*16 IFSC,d1_IFSC,D_IF

C *****************************************************************


C INTERNAL VARIABLES
integer size,iflag
      complex*16 smc1inv(7,7),workc1(7,14)

      size = 7
      iflag = 0


      CALL MINV(smc1,smc1inv,workc1,size,iflag)
```

```
C CIRCUMFERENTIAL      EXCITATION   exctype = 2
C RADIAL               EXCITATION   exctype = 1
C AXIAL                EXCITATION   exctype = 0

      if (exctype .eq. 1) then
         A1_C1 = -smc1inv(1,1)
         A2_C1 = -smc1inv(2,1)
         B1_C1 = -smc1inv(3,1)
         B2_C1 = -smc1inv(4,1)
         C1_C1 = -smc1inv(5,1)
         C2_C1 = -smc1inv(6,1)
         M_OF = (A1_C1*d1_SP_CY1_b1 +
     1         A2_C1*d1_SP_CY1_b2 +
     1         B1_C1*n/b*VXP_CY1_b1 +
     1         B2_C1*n/b*VXP_CY1_b2 +
     1         C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
     1         C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
         D_IF = -smc1inv(7,1)
      elseif (exctype .eq. 0) then
         A1_C1 = -smc1inv(1,2)
         A2_C1 = -smc1inv(2,2)
         B1_C1 = -smc1inv(3,2)
         B2_C1 = -smc1inv(4,2)
         C1_C1 = -smc1inv(5,2)
         C2_C1 = -smc1inv(6,2)
         M_OF = (A1_C1*d1_SP_CY1_b1 +
     1         A2_C1*d1_SP_CY1_b2 +
     1         B1_C1*n/b*VXP_CY1_b1 +
     1         B2_C1*n/b*VXP_CY1_b2 +
     1         C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
     1         C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
         D_IF = -smc1inv(7,2)
      elseif (exctype .eq. 2) then
         A1_C1 = -smc1inv(1,3)
         A2_C1 = -smc1inv(2,3)
         B1_C1 = -smc1inv(3,3)
         B2_C1 = -smc1inv(4,3)
         C1_C1 = -smc1inv(5,3)
         C2_C1 = -smc1inv(6,3)
         M_OF = (A1_C1*d1_SP_CY1_b1 +
     1         A2_C1*d1_SP_CY1_b2 +
     1         B1_C1*n/b*VXP_CY1_b1 +
     1         B2_C1*n/b*VXP_CY1_b2 +
     1         C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
     1         C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
         D_IF = -smc1inv(7,3)
```

```
endif


return
end
```

```
SUBROUTINE ABC_RC1_INVERT(n,exctype,smrc1,b)

C EXTERNAL VARIABLES

      integer n,exctype
      real*8 b
      complex*16 smrc1(10,10)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
     1            SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
     1            SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
     1            SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
     1            VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
     1            VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
     1            VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
     1            VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
     1            VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
     1            VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
     1            VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
     1            VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
     1            lame_c1,shear_c1,cl_c1,ct_c1,
1          A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C *****************************************************************
```

```
C DEFINITIONS FOR COMMON BLOCK /ROD/

      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod


      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C ***************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      complex*16 OFSC,d1_OFSC,M_OF

      common /OFLUID/ OFSC,d1_OFSC,M_OF

C ***************************************************************

C INTERNAL VARIABLES

      integer size,iflag
      complex*16 smrc1inv(10,10),workrc1(10,20)

      size = 10
      iflag = 0


      CALL MINV(smrc1,smrc1inv,workrc1,size,iflag)
```

```
C CIRCUMFERENTIAL     EXCITATION exctype = 2
C RADIAL              EXCITATION exctype = 1
C AXIAL               EXCITATION exctype = 0

      if (exctype .eq. 1) then
         A1_C1 = -smrc1inv(1,1)
         A2_C1 = -smrc1inv(2,1)
         B1_C1 = -smrc1inv(3,1)
         B2_C1 = -smrc1inv(4,1)
         C1_C1 = -smrc1inv(5,1)
         C2_C1 = -smrc1inv(6,1)
        A1_rod = -smrc1inv(7,1)
        B1_rod = -smrc1inv(8,1)
        C1_rod = -smrc1inv(9,1)
          M_OF = -smrc1inv(10,1)
      elseif (exctype .eq. 0) then
         A1_C1 = -smrc1inv(1,2)
         A2_C1 = -smrc1inv(2,2)
         B1_C1 = -smrc1inv(3,2)
         B2_C1 = -smrc1inv(4,2)
         C1_C1 = -smrc1inv(5,2)
         C2_C1 = -smrc1inv(6,2)
        A1_rod = -smrc1inv(7,2)
        B1_rod = -smrc1inv(8,2)
        C1_rod = -smrc1inv(9,2)
          M_OF = -smrc1inv(10,2)
      elseif (exctype .eq. 2) then
         A1_C1 = -smrc1inv(1,3)
         A2_C1 = -smrc1inv(2,3)
         B1_C1 = -smrc1inv(3,3)
         B2_C1 = -smrc1inv(4,3)
         C1_C1 = -smrc1inv(5,3)
         C2_C1 = -smrc1inv(6,3)
        A1_rod = -smrc1inv(7,3)
        B1_rod = -smrc1inv(8,3)
        C1_rod = -smrc1inv(9,3)
          M_OF = -smrc1inv(10,3)
      endif



      return
      end
```

```
SUBROUTINE ABC_RC2_INVERT(n,exctype,smrc2,a,c)

C EXTERNAL VARIABLES

      integer n,exctype
      real*8 a,c
      complex*16 smrc2(13,13)

C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2

      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
   1              SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
   1              SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
   1              SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
   1              VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
   1              VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
   1              VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
   1              VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
   1              VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
   1              VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
   1              VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
   1              VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
   1              lame_c1,shear_c1,cl_c1,ct_c1,
   1              A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

C ******************************************************************
```

```
C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

      complex*16 lame_c2,shear_c2,cl_c2,ct_c2

      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2


common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
   1           SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
   1           SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
   1           SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
   1           VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
   1           VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
   1           VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
   1           VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
   1           VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
   1           VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
   1           VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
   1           VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
   1           lame_c2,shear_c2,cl_c2,ct_c2,
   1           A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

C ******************************************************************


C DEFINITIONS FOR COMMON BLOCK /ROD/

       common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
```

```
     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

      complex*16 OFSC,d1_OFSC,M_OF

      common /OFLUID/ OFSC,d1_OFSC,M_OF

C *****************************************************************

C *****************************************************************

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

      common /IFLUID/ IFSC,d1_IFSC,D_IF

      complex*16 IFSC,d1_IFSC,D_IF

C *****************************************************************

C INTERNAL VARIABLES

      integer size,iflag
      complex*16 smrc2inv(13,13),workrc2(13,26)

      size = 13
      iflag = 0

      CALL MINV(smrc2,smrc2inv,workrc2,size,iflag)
```

```
C CIRCUMFERENTIAL     EXCITATION exctype = 2
C RADIAL              EXCITATION exctype = 1
C AXIAL               EXCITATION exctype = 0

      if (exctype .eq. 1) then
        A1_C2 = -smrc2inv(1,1)
        A2_C2 = -smrc2inv(2,1)
        B1_C2 = -smrc2inv(3,1)
        B2_C2 = -smrc2inv(4,1)
        C1_C2 = -smrc2inv(5,1)
        C2_C2 = -smrc2inv(6,1)
        A1_C1 = -smrc2inv(7,1)
        A2_C1 = -smrc2inv(8,1)
        B1_C1 = -smrc2inv(9,1)
        B2_C1 = -smrc2inv(10,1)
        C1_C1 = -smrc2inv(11,1)
        C2_C1 = -smrc2inv(12,1)
        D_IF = -smrc2inv(13,1)
        M_OF =(A1_C2*d1_SP_CY2_c1 +
     1          A2_C2*d1_SP_CY2_c2 +
     1          B1_C2*n/c*VXP_CY2_c1 +
     1          B2_C2*n/c*VXP_CY2_c2 +
     1          C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
     1          C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

      elseif (exctype .eq. 0) then
        A1_C2 = -smrc2inv(1,2)
        A2_C2 = -smrc2inv(2,2)
        B1_C2 = -smrc2inv(3,2)
        B2_C2 = -smrc2inv(4,2)
        C1_C2 = -smrc2inv(5,2)
        C2_C2 = -smrc2inv(6,2)
        A1_C1 = -smrc2inv(7,2)
        A2_C1 = -smrc2inv(8,2)
        B1_C1 = -smrc2inv(9,2)
        B2_C1 = -smrc2inv(10,2)
        C1_C1 = -smrc2inv(11,2)
        C2_C1 = -smrc2inv(12,2)
        D_IF = -smrc2inv(13,2)
        M_OF =(A1_C2*d1_SP_CY2_c1 +
     1          A2_C2*d1_SP_CY2_c2 +
     1          B1_C2*n/c*VXP_CY2_c1 +
     1          B2_C2*n/c*VXP_CY2_c2 +
     1          C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
     1          C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC
```

```
      elseif (exctype .eq. 2) then
        A1_C2 = -smrc2inv(1,3)
        A2_C2 = -smrc2inv(2,3)
        B1_C2 = -smrc2inv(3,3)
        B2_C2 = -smrc2inv(4,3)
        C1_C2 = -smrc2inv(5,3)
        C2_C2 = -smrc2inv(6,3)
        A1_C1 = -smrc2inv(7,3)
        A2_C1 = -smrc2inv(8,3)
        B1_C1 = -smrc2inv(9,3)
        B2_C1 = -smrc2inv(10,3)
        C1_C1 = -smrc2inv(11,3)
        C2_C1 = -smrc2inv(12,3)
        D_IF = -smrc2inv(13,3)
        M_OF =(A1_C2*d1_SP_CY2_c1 +
     1          A2_C2*d1_SP_CY2_c2 +
     1          B1_C2*n/c*VXP_CY2_c1 +
     1          B2_C2*n/c*VXP_CY2_c2 +
     1          C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
     1          C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

      endif


      return
      end
```

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Defense Technical Information Center	12
Office of Naval Research (Code 321SS: K. Dial, S. Littlefield, R. Varley)	3
Naval Research Laboratory (A. Dandridge, S. Vohra)	2
Cambridge Acoustical Associates (J. Garrelick, J. Cole III)	2
Applied Measurement Systems, Inc., New London (J. Diggs)	2